Full name: Elsufev Aleksandr Mikhailovich
Address: refugee from Russia, now in Tbilisi, Georgia
Phone: 995-558-13-38-77 (Georgian)
Telegram: @Elsufiev_al2
Email: Elsufiev_al@mail.ru, a.m.elsufev@gmail.com
LinkedIn: https://www.linkedin.com/in/aleksandr-elsufev-2b86a9305/

Target

I am applying for the position of programmer-analyst, where I can apply my extensive knowledge in the field of working with databases, data analysis, programming, mathematics, mathematical modeling and neural network forecasting.

Education

St. Petersburg State University

Faculty of Mathematics and Mechanics, Department of Probability Theory and Mathematical Statistics

BSc (Hons), 1997

FMS-30, 1992

Skills:

- Programming languages:

SAS, SQL – perfect, over 20 years of work, with advanced training courses.

R – at a good level, 3 years of intensive work, self-training.

Python - studying now.

• Databases: DB2, PostgreSQL

• Data Analysis and Visualization: Experience working with large volumes of data, identifying anomalies and trends.

• Neural network forecasting: Development, creation and implementation of forecast models.

• Optimization and algorithms: Development of optimization algorithms for business processes.

Experience

RZD (Russian Railways) 1999-2023:

• Information and Computing Center of Oktyabrskaya Railway (RZD) 1999-2014, programmer.

• Trans-IT (RZD) 2015-2016, leading programmer, specialist in mathematical software.

• **VNIIZHT** (All-Russian Scientific Research Institute of Railway Transport). (RZD) 2016-2023, leading programmer, specialist in mathematical software.

Achievements

• Development and maintenance of software based on SAS (v.4, v.6, v.8), working with databases on DB2, excellent knowledge of SQL.

• Interaction with the head office of SAS on issues of bug fixes and software updates.

• Programming and maintenance of SAS-based information systems for processing primary information and issuing final analytical, statistical and forecast reports for Russian Railways management.

• Development, writing and implementation of an operational forecasting system for suburban transportation of OktZhD based on SAS.

• Creation of a neural network forecasting complex for passenger flows of the entire Russian Railways in the R language. Invention of a new algorithm for predicting a quantity that has a natural limiter.

• Mathematical theoretical study of passenger flows, and as a result, optimization of the process of data analysis and ticket pricing, which led to significant financial savings for the company.

• Development of an algorithm for optimizing the operation of the locomotive fleet, and writing software to reduce operating costs.

• Identification of mass thefts on the Sakhalin Railway by analyzing primary data on ticket sales.

• Participation in the project for the development of the Express-3 ticket sales system to the new version of Express-NG (new generation). Proposal and implementation of methods for mathematical data processing, which made it possible to significantly increase the performance of the entire complex.

• Execution of one-time assignments from management for data analysis that are not related to the main activity, but require deep knowledge of mathematics and data analysis skills. On the results of the work performed I once made a report at a meeting of Russian Railways specialists.

• Regular assistance to colleagues in improving their software and finding errors in software, mentoring on SQL issues.

• Currently learning Python

Additional skills and personal qualities

• Experience in teaching mathematics and teaching practice.

• Multiple successful participation in olympiads in mathematics and physics, including All-Union (USSR) and All-Russian.

• Ability to work in a team and provide support to colleagues.

• Creative approach to solving complex problems and analytical thinking.

• Experience in helping colleagues find logical errors in other people's software.

• Extensive experience in solving mathematical puzzles and highly complex problems on a PC.

• Long-term experience of semi-professional volunteering in massage and bard music.

• I am interested in sports and athletics, in 2022 I received 1st category of the GTO.

• I am interested in all areas of science. I create my own theory of the origin of stars, which later turned out to be a development of the theory of astrophysicist Carl Rouse, and wrote a book on it.

I will provide a portfolio upon request.

Thank you in advance for considering my candidacy.
Sincerely,
Elsufev Aleksandr

ФИО: Елсуфьев Александр Михайлович
Адрес: беженец из России, сейчас в Тбилиси, Грузия
Телефон: 995-558-13-38-77 (грузинский)
Телеграм: @Elsufiev_al2
Электронная почта: Elsufiev_al@mail.ru , a.m.elsufev@gmail.com
ЛинкедИн: https://www.linkedin.com/in/aleksandr-elsufev-2b86a9305/

Цель

Претендую на позицию программиста-аналитика, где могу применить свои обширные знания в области работы с базами данных, анализа данных, программирования, математики, математического моделирования и нейросетевого прогнозирования.

Образование

Санкт-Петербургский Государственный Университет
Факультет математики и механики, Кафедра теории вероятностей и математической статистики
Бакалавр наук (с отличием), 1997

ФМШ-30, 1992г

Навыки:

- **Языки программирования**:

SAS, SQL – в совершенстве, свыше 20 лет работы, с прохождением курсов повышения квалификации.
R – на хорошем уровне, 3 года интенсивной работы, самообучение.
Python -изучаю сейчас.

- **Базы данных**: DB2, PostgreSQL
- **Анализ данных и визуализация**: Опыт работы с большими объемами данных, выявление аномалий и тенденций.
- **Нейросетевое прогнозирование**: Разработка, создание и внедрение прогнозных моделей.

- **Оптимизация и алгоритмы**: Разработка алгоритмов оптимизации для бизнес-процессов.

Опыт работы

РЖД (Российские железные дороги) 1999-2023г:
- ИВЦ Октябрьской ЖД (РЖД) 1999-2014г, программист.
- Транс-ИТ (РЖД) 2015-2016г, ведущий программист специалист по математическому обеспечению.
- ВНИИЖТ (РЖД) 2016-2023Г, ведущий программист специалист по математическому обеспечению.

Достижения

- Разработка и сопровождение программного обеспечения на базе SAS (v.4, v.6, v.8), работа с базами данных на DB2, отличное знание SQL.
- Взаимодействие с головным офисом компании SAS по вопросам исправления ошибок и обновления ПО.
- Программирование и сопровождение информационных комплексов на базе SAS, для обработки первичной информации и выдачи итоговой аналитической, статистической и прогнозной отчетности для руководства РЖД.
- Разработка, написание и внедрение системы оперативного прогнозирования пригородных перевозок ОктЖД на базе SAS.
- Создание комплекса нейросетевого прогнозирования пассажиропотоков всей РЖД на языке R. Изобретение нового алгоритма прогнозирования величины, обладающей естественным ограничителем.
- Математическое теоретическое исследование пассажиропотоков, и по итогу, оптимизация процесса анализа данных и ценообразования билетов, что привело к значительной финансовой экономии компании.
- Разработка алгоритма оптимизации работ локомотивного парка, и написание ПО, для снижения эксплуатационных затрат.
- Выявление массовых хищений на Сахалинской железной дороге путём анализа первичных данных о продаже билетов.
- Участие в проекте развития системы продажи билетов «Экспресс-3» до новой версии «Экспресс-НП» (новое поколение). Предложение и реализация методов математической обработки данных, позволивших существенно повысить быстродействие всего комплекса.
- Исполнение разовых поручений руководства по анализу данных, не относящихся к основной деятельности, но требующих глубоких знаний математики и навыков анализа данных. По результатам выполненной работы однажды делал доклад на совещании специалистов РЖД.

- Регулярная помощь коллегам в улучшении их ПО и поиске ошибок в ПО, наставничество по вопросам SQL.
- Сейчас изучаю язык Python

Дополнительные навыки и личные качества

- Опыт преподавания математики и педагогическая практика.
- Многократное успешное участие в олимпиадах по математике и физике, включая всесоюзные (СССР) и всероссийские.
- Умение работать в команде и оказывать поддержку коллегам.
- Творческий подход к решению сложных задач и аналитическое мышление.
- Опыт помощи коллегам в поиске логических ошибок в чужом ПО.
- Большой опыт решения математических головоломок и задач высокой сложности на ПК.
- Длительный опыт полупрофессионального волонтёрства в массаже и бардовской музыке.
- Увлекаюсь спортом, лёгкой атлетикой, в 2022г получил 1 разряд ГТО.
- Интересуюсь всеми областями в науке. Создал свою теорию происхождения звёзд, впоследствии оказавшуюся развитием теории астрофизика Карла Роуза, и по ней написал книгу.

Портфолио предоставлю по запросу.

Заранее благодарю за рассмотрение моей кандидатуры.

С уважением,
Елсуфьев Александр

Portfolio

INTRODUCTION. Briefly, how I differ from all other candidates.

1. Achievements at school and university
2. Main work on suburban and passenger transportation databases.
3. Processing of data on employee dismissals
4. Processing data on server load and queues
5. Optimization of pricing for trains
6. Simplifying work with train schedules
7. Optimization of locomotive operations
8. Neural network forecasting
9. Detection of fraud on the Sakhalin Railway
10. Interview at Gazprom for the position of programmer-mathematician
11. Writing a book on astrophysics

INTRODUCTION. Briefly, how I differ from all other candidates.

1. Education and Career: I graduated from St. Petersburg State University (SPbSU) with honors in mathematics in 1997. For almost 25 years I worked at Russian Railways, starting as a junior programmer and moving up to a leading mathematical programmer. My experience includes helping colleagues solve current problems, find logical errors and optimize programs. I am always ready to provide such assistance in a new company, after the internship period.

2. Data analysis: among other things I identified a fraud scheme in Russian Railways worth hundreds of millions of rubles by analyzing the database of tickets sold on Sakhalin.

3. Math Seminars: I conducted math seminars for colleagues. In preparation for one of the seminars, I proved a theorem about the criterion for the optimal relationship between ticket price and other factors, which was introduced into the Russian Railways software package.

4. Forecasting: I developed a new principle of neural network forecasting of a random variable with a natural limiter, which increased the correlation of the forecast with the original data from 25% to 40%. I consider this solution to be my personal invention.

5. Program optimization: I helped colleagues optimize their programs, including the successful optimization of the Express-6 system, which made it possible to perform calculations on a regular PC instead of SAP HANA servers.

6. Mathematical modeling and probability theory: I saved the IOMM system from problems on the server, using the theory of queues, and on my personal initiative developed a complex for optimizing the work of the Russian Railways locomotive fleet, which could provide significant savings.

7. Knowledge of physics: I have deep knowledge of physics, especially mechanics. In response to my child's question, I developed a theory of the origin of stars that answers many questions left unsatisfactorily answered by the modern astrophysical community.

1 Achievements at school and university

While studying at school, at Physics School-30, I repeatedly participated in city olympiads in mathematics, physics, chemistry and programming. I twice received a 3rd degree diploma in programming, nothing in chemistry, 3 3rd degree diplomas in physics, and 1st degree in 11th grade with a recommendation to go to the All-Union, but I had to refuse because of mathematics.

In mathematics at school I twice received a 1st degree diploma in the city, in the 10th and 11th grade, after which I went to the All-Union Olympiads in Smolensk and Alma-Ata, both times receiving a 3rd degree diploma.

At the university, in my 1st year, I took part in the informal internal MatMekhov Olympiad, took 3rd place among students of all courses, and was officially recognized as the best first-year student, with a prize. From the 2nd semester of the 1st year, as part of a group of the best MathMech students, he was transferred to the LOMI group (half of the lectures were listened to directly at the former LOMI, now POMI RAS named after Steklov, and exams were taken there), which existed until the end of the 2nd year.

In the 2nd and 3rd years, I participated in city student olympiads in mathematics, won 1st place, after which, in my 3rd year, I went to the All-Union Olympiad in Chelyabinsk, where I received 3rd place (a distinctive feature of this Olympiad was that there was only 1st place, only 2nd place, and only 3rd place in all over Russia, all 3 places were taken by our group MatMekhovtsev).

In my 3rd year I participated in the student Olympiad in mechanics, and was the only student from MatMech sent to the city Olympiad held at the Polytechnic University. There I took 1st place, solving all the problems, but since the Olympiad turned out to be a team competition (5 people from each university) and I was alone, I was counted out of the competition.

I wrote my diploma thesis at the Department of Probability Theory and Mathematical Statistics, on the topic “behavior of type 2 P-values under alternative conditions”, the work was eventually published in the final journal of the POMI RAS in 1997. I graduated from the university with honors, with scores of 3-4th diploma in the entire Mathmekh.

In the 1st-4th years of the university, I helped lead a mathematics club at PhM-239. In the 5th year in 1996, he underwent teaching practice at the University of Finance and Economics, taught practical work in mathematics, as a result, the group led by another showed an average score of over 4.5 in the exam, in three other groups the maximum was 3.7.

2 Major work on commuter and passenger transportation databases.

The first main work at Russian Railways was to create an aggregated monthly and daily reporting database from the database of tickets sold (suburban and passenger) with the aim of issuing certificates and reports of several different types for higher management, for further optimization of the transportation process.

Basic parameters of input data. For the suburbs, the OktZhD ticket sales database is a monthly updated database, with a volume from 50 thousand lines (1998) to 3 million lines (2023) per month, gradually increasing detail, indefinite storage. For passenger transportation - a database of up to 1 million rows per day, a storage period of 2 years, there was a need for the simplest aggregated data, and for many registers of different beneficiaries.

The initial suburban data consisted of 3 large tables - one for all tickets sold, and the other two - for each of the tickets sold, the route of movement of each specific passenger is indicated, and the division of the cost of his travel by region (the cost of the ticket consists of the costs by region, assigned by agreement with these regions), and were almost unsuitable for issuing serious reports. Additional tables were 3-5 times longer than the main ticket sales table.

To speed up and simplify the work, the first step was to normalize the data - there was only one table of the data itself, and several directories of unique options for the type of ticket, the place of sale, and the route itself (including a breakdown of the cost by region). As a result, the longest cumulative lookup table (routes) over 25 years of operation had 2-3 times fewer data rows than the raw data that was written to it in 1 month. The second result of normalization was memory savings, which seriously affected further deep data processing.

As a result, a structure was created for the suburb of several types of aggregates, for fundamentally different types of reports, and reference tables for them. The main need of the SZPPK was for certificates such as “who sold how many tickets of different types, or transported passengers, in different periods of time, and comparison with previous periods.” For these purposes, a structure was developed consisting of several reference tables and aggregate tables with approximately the following structure:

-the reporting month

-station (department, road, administrative district) who performed the action.

-type of ticket (single or subscription, discount, payment method, validity period).

-action performed (name = selling a ticket, receiving money, calculating the amount of losses, sending a passenger or passenger turnover, and much more)

- the calculated volume itself (tickets, revenue, losses, generated or dispatched passengers or passenger turnover)

-date and account number

As a result, the aggregated data table had a volume of about 100 thousand rows per month, the execution of any query fit into the 0.2 seconds required by the system, sometimes increasing to 1 second when the server was heavily loaded.

To optimize the creation of SQL queries for different references, one universal query was written that returned all possible parameters in all possible sections that were required for reporting (sometimes the query had to be replenished when new requirements from the customer appeared), after which a specially written program “cut off everything "superfluous" from it, leaving only those parameters that are needed in this specific help - and it is precisely this request that has already been included in the listing of a specific program that issues a specific help. In fact, the SAP HANA system, which I got to know at Trans-IT in 2015-2017, does the same thing, only it worked for us already in 2007.

Thanks to the proposed structure, which contains the date and calculation number, the process of finding various software errors, which had to be regularly modified to meet new requirements, was greatly simplified. When changing the software and recalculating aggregates for previous months, the program first calculates the full volume for each indicator, then subtracts the volumes already recorded in the database as a result of previous calculations, after which it throws out all lines with zero data, and writes only the result of the change to the aggregates, with the corresponding date and calculation number. For this reason, after recalculating the data, it was always possible to see what and how much had changed, and if it had changed in the wrong way, or not as much as expected, roll back and adjust the software again.

In addition to these main units, there were also units for type reports - “how many passengers traveled along the route from station A to station B”, “what was the daily departure of passengers from each station, and how the average travel distance changed” (the original data came once every month, but starting from 2010, already broken down by the start date of the ticket) - similar aggregates were made for them, with a slightly different structure, in order to minimize the volumes in each aggregate table.

The most complex table was created for certificates such as “what was the passenger turnover, and how many unique passengers traveled during the reporting period in different regions of the OktZhD network” - keeping in mind that the certificate was used by several companies (SZPPK, MTPPC, DOSS), and each of them wanted see the same data in different drillings, the description of which fit into a separate reference book that had a recurrent link structure. The certificate was issued in the form {value for the road, its division into the required areas {value for the area, its division into subdistricts {...}}} - and so from the data along the entire road there was a gradual refinement of the data to each smallest stage (= railway section between two neighboring stations). The mathematical complexity

of this reporting was that, among other things, it was necessary to indicate unique passing passengers, that is, if a passenger traveled from A to B through 20 stages, then in each of them he had to be counted 1 time, and in any region at least somehow affected his route of movement - it was also counted 1 time, and not by the number of stages included in the section in question. That is, the program had to produce a value for each region that was calculated not by summation, and not on the basis of raw data, but by very small aggregates and a directory arbitrarily specified by the user. As a result, the software for issuing certificates was based on the theory of flat graphs, while the units contained data only for each section (there are only about 1,500 of them on OktZhD), without information about any larger entities, due to which the certificate was processed in a matter of seconds.

Similar aggregated data bases were built on passenger traffic, but they were in little demand due to the lack of private companies working on them, and the management of the centralized Russian Railways did not optimize its processes.

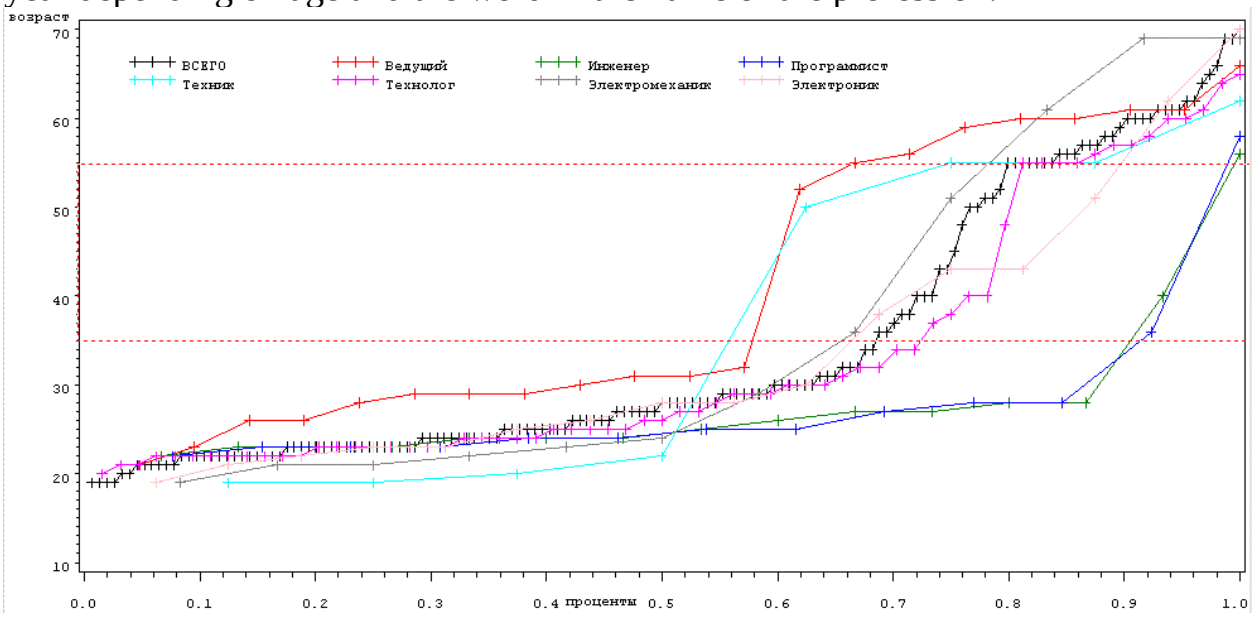
The entire complex of calculations of data on suburban transportation for OktazhD was written by me without outside help in the SAS language, with a connection to the storage of source and aggregated data on a DB2 server, and with the issuance of references for SQL queries in Excel forms - I carried out all the work in this system before drawing Excel reports, which were already done by other programmers.

It was planned to create approximately the same type of database in the still-developed Express-NP system, which would issue reports to the entire Russian Railways network.

3 Processing of data on employee dismissals

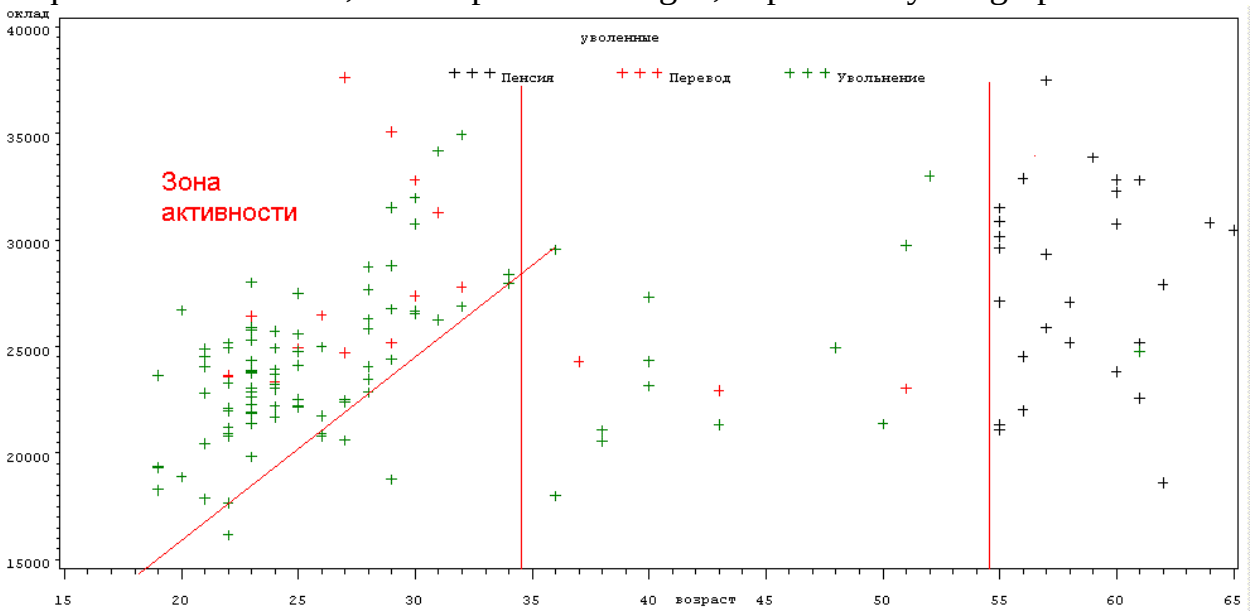
In 2012, while still working at the OktZhD IEC, a request was received from the HR department to find out the reasons and criteria for dismissal of Russian Railways employees, in order to further optimize work with personnel. The input was a database of all Russian Railways employees with numerous parameters at that time, and a separate list of all those who quit over the last 15 years, with their indicators at the time of dismissal.

After checking all the data, eliminating dismissals due to transfer to another position, death or retirement, checking all the data on dismissals could not immediately reveal anything, no matter what sections the data was reviewed. A typical example is a graph of the probability of an employee quitting during the year depending on age and the word in the name of the profession.



But then an approximate normalization of the data was carried out, bringing them to the current state, namely, over the period of history under review, salaries increased more than 10 times, and therefore the salaries of those who quit and those who remained were brought to a single value, namely, multiplied by an exponential with the indicator inflation and the time from dismissal to the moment

in question. As a result, a clear pattern emerged, expressed by the graph



From this graph it became clear that people retire mainly due to age (but there are others, for example, due to disability), and for transfer and leaving Russian Railways for good, it is subject to a clear criterion - only if the employee has a good specialty and the opportunity for growth (= salary above a linear function of age, and age up to 35 years, that is, a young specialist), then he often quits, regardless of all other indicators (of course, salary correlates with profession). Otherwise, if an employee is not competitive in the labor market, his dismissal rate is minimal.

In this form, the answer to the question posed was more than good for the HR department, and the problem was considered solved. I don’t know what managerial conclusions were drawn from this.

(Illustrations taken from the surviving file of the presentation of the final work to superiors)

4 Processing data on server load and queues

Around 2007, an assignment was received - to analyze the cause of problems with data entry into the IOMM (Integrated Processing of Drivers Routes) system, which resulted in the emergence of long queues to the data processing server for processing requests for data entry of drivers' daily work.

That is, to record and control the work of the drivers, dispatchers manually entered data about the driver's work on local computers, pressed "enter", the task was sent to the server, and there it was in the queue until it was executed, sometimes for tens of minutes. At the same time, all the server operation logs indicated that each specific route is normally processed within 1-2 seconds, and during peak hours – up to 10 seconds, but the queues did not allow the entire OktZhD to work normally.

To solve the problem, server logs were collected, showing what percentage of the load it had at each moment in time, when which program was launched, and specifically for IOMM for some requests - at what time the request was sent to the server (the log itself recorded only the moment of the start and end of processing one route, not taking into account the time it spent in queue).

To predict the waiting time for a request in a queue, a branch of mathematics was used - queuing theory. And as a result, a program was written that predicted with great accuracy exactly when a request received for processing would reach its turn and be processed. The program used only 2 parameters that were easily calculated from the log table - how much of the last 10 minutes the server was processing requests, and how many requests were processed as a result.

Data processing revealed that the server never operated at more than 50% of its capacity, queues formed only at the level of 25 or 50% of the server load, but very often there were no queues at this load level. A more detailed examination of the periods when queues occur showed that it occurs only when the server is very busy with two programs out of about 20 constantly running on it, and almost does not occur with other programs.

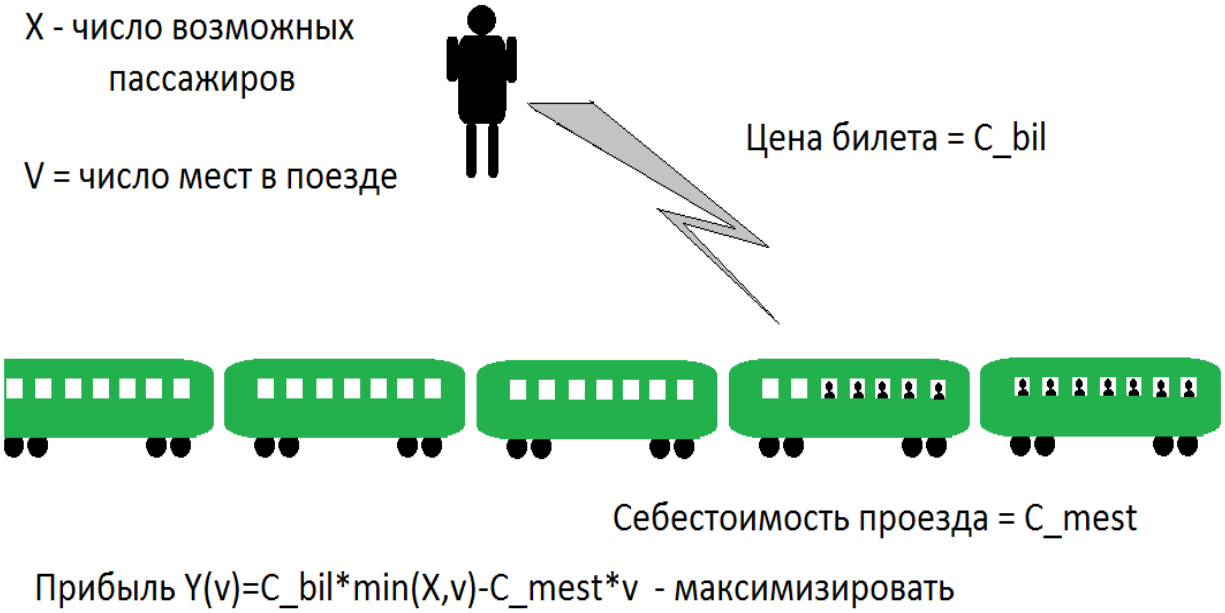
As a result, it was suggested that the server is a 4-processor computer in which different programs are executed on different processors, always on the same ones. When this assumption was conveyed to the administrators, it turned out that this was the latest single-processor server with 4 cores, performing exactly as predicted. And since no one assigned any priority or location to the programs, they divided the "areas of responsibility" according to the dates the programs were installed on the server, each running strictly on its own single core, and without loading all neighboring cores. At the same time, one of the four cores never performed any useful work at all.

After correcting this defect, the IOMM program worked without queues, executing requests “on the fly”; the work of all other programs also improved.

The result of the work was an offer for me to give my report at a rally of young OktZhD specialists, held in the fall of 2008 at the Pribaltiyskaya Hotel, after which, according to unverified information, my report was published in the Russian Railways Bulletin. Another result of this work was my assignment to the category of leading programmer, with an increase in rank according to the Russian Railways tariff schedule.

5 Optimization of pricing for trains

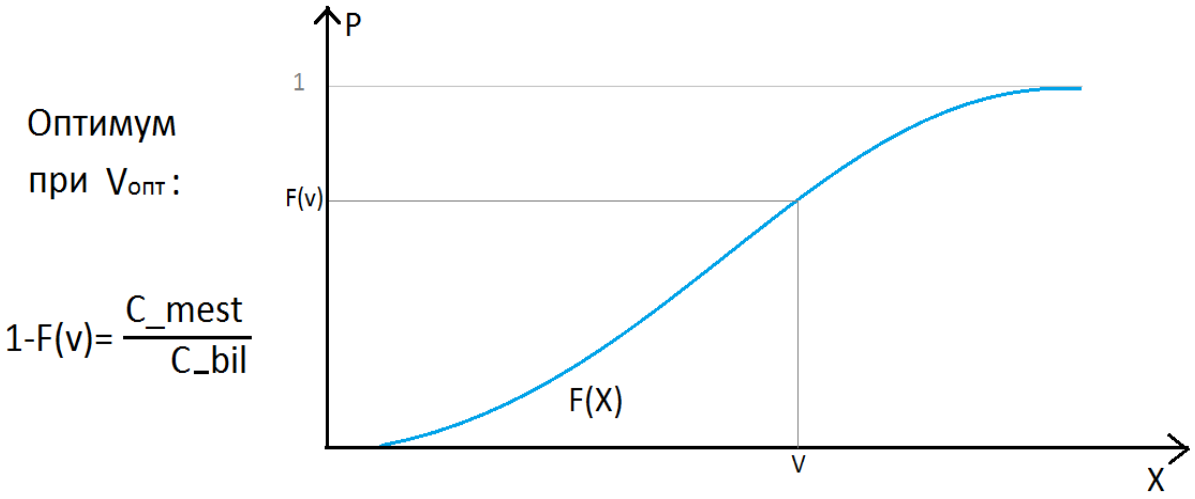
In 2013, it was received from the management of the OktZhD Information Computing Center to hold a seminar on the theory of probability and its possible application in the work of the Information Computing Center, with a free choice of examples. As one of the possible examples, I chose the problem - how to maximize the possible profit of a carrier launching a train with V free seats, from which X pieces will be purchased by passengers - which is a random variable with an unknown distribution. The carrier has the right to change the number of seats, as well as (not always) the cost of tickets (very often the cost of travel is regulated by the state).



Прибыль Y(v)=C_bil*min(X,v)-C_mest*v - максимизировать

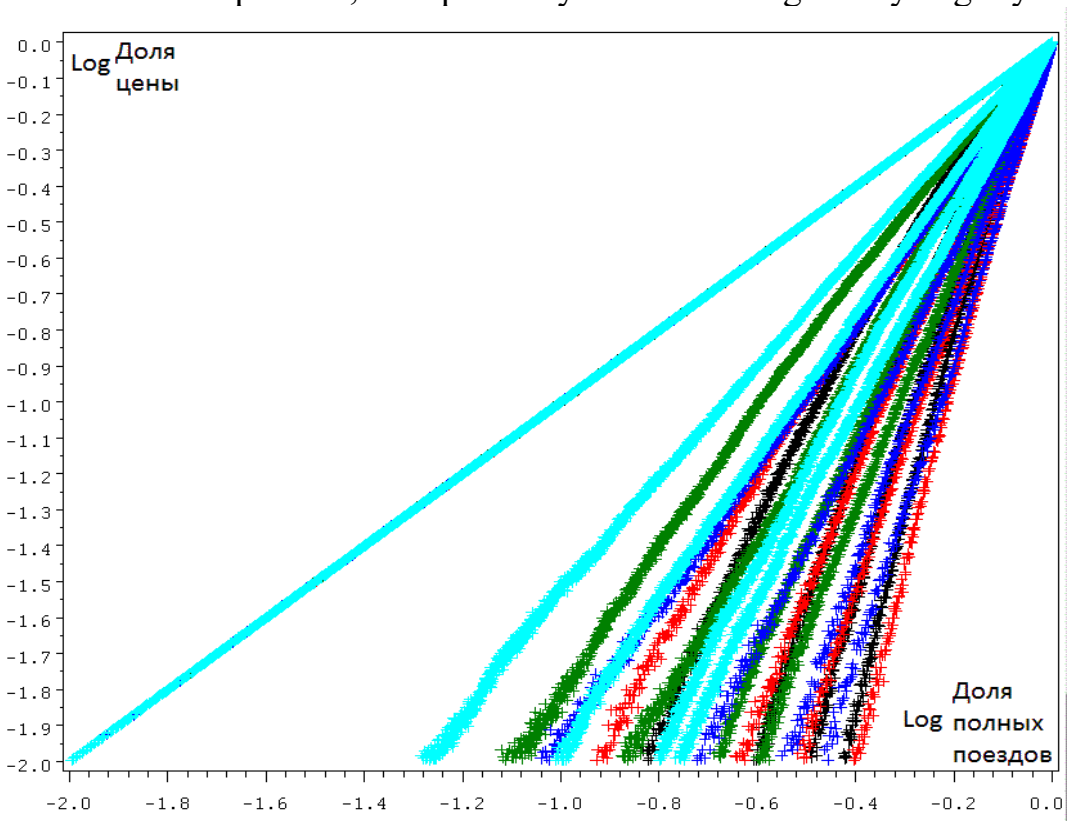
It is hardly possible to solve such a problem in its entirety, but there are special cases - options for purchasing tickets without deferred demand and competition, that is, when passengers need to travel “exactly today and only on this train,” and tomorrow it will be too late. Under this assumption, a mathematical calculation was carried out, and it turned out that the criterion for the optimality of the offer of the number of places is amazingly simple and easy to calculate. Namely, the optimal number of seats offered, or their selling prices, will be when the probability that the train will be packed (100% of the seats are purchased) is exactly equal to the ratio of the cost of the ticket to the proposed retail sale price. The formula can also take into account the cost of car downtime - when it is non-

zero, for example, due to the need to service cars standing in the settling tank.



$$\text{Вероятность 100\% заполнения поезда} = \frac{\text{Себестоимость проезда - Стоимость простоя}}{\text{Стоимость билета}}$$

Further research using mathematical modeling showed that with the widest range of passenger distribution functions, and with different criteria for deferred demand and competition, the optimality formula changes very slightly.



The result of voicing this work was that the passenger company DOSS ordered another column in all its passenger reporting certificates for long periods - “the number of trains sent completely full,” meaning by this the sale of over 95%

of the tickets offered for sale, and, of course, the share fully filled trains among all dispatched ones. Apparently, this optimizing factor is still used today.

(Illustrations taken from the surviving file of the presentation of the final work to superiors)

6 Simplifying work with train schedules

In 2016, during a short period of work at the Trans-IT company, colleagues from a neighboring department had a serious problem with server capacity, although they were then working on SAP HANA, with the most powerful server possible at that time.

The situation was like this. Initially, Russian Railways had a train schedule - about 3,000 regular trains running on approximately constant routes with an approximately constant schedule, but with numerous minor differences. In the old system, schedules were controlled by many files, where each train was assigned which of the possible schedules it would follow on any given day. In the new system, all this had to be in the database, and therefore, since the database had to be uniform, the table was created something like this

-Train number (about 3000 pieces)

-departure date (from -50 to +360 days from the current date)

-route station with all its parameters (from 20 to 1000 lines of recording, depending on the length of the route).

As a result, the resulting table had over 100 million rows of records, which created a simple structure with simple addressing of requests, but due to its size and the requirements for constant (thousands of times per minute) gluing it with incoming data requests, it simply collapsed the entire system. The result was paralysis of the development process of the new Express-5 system on SAP HANA.

When they asked me what could be done about this, I proposed and partially independently implemented the following algorithm. The database of all schedules was divided into 2 separate tables. The first table - unique schedules - when entering a new schedule for a specific train on a specific date into the database, it is checked whether there has been such a unique schedule before; if it was found, its unique number was taken, if not, it was entered with a new number, and that was taken. The second table shows the correspondence of a specific train number with a specific departure date to a unique schedule number. As a result, the entire table of unique schedules contained only about 1 million rows, and the table of train departure dates was even smaller, and such a combination worked like a charm, almost without loading the processor.

The most difficult algorithmic part was figuring out how to find out whether the new train route schedule is already known in the table of all unique schedules, or whether it needs to be added there? The fact is that if the new schedule has 1000 lines, and the table of unique ones is 1 million, then direct gluing with all the

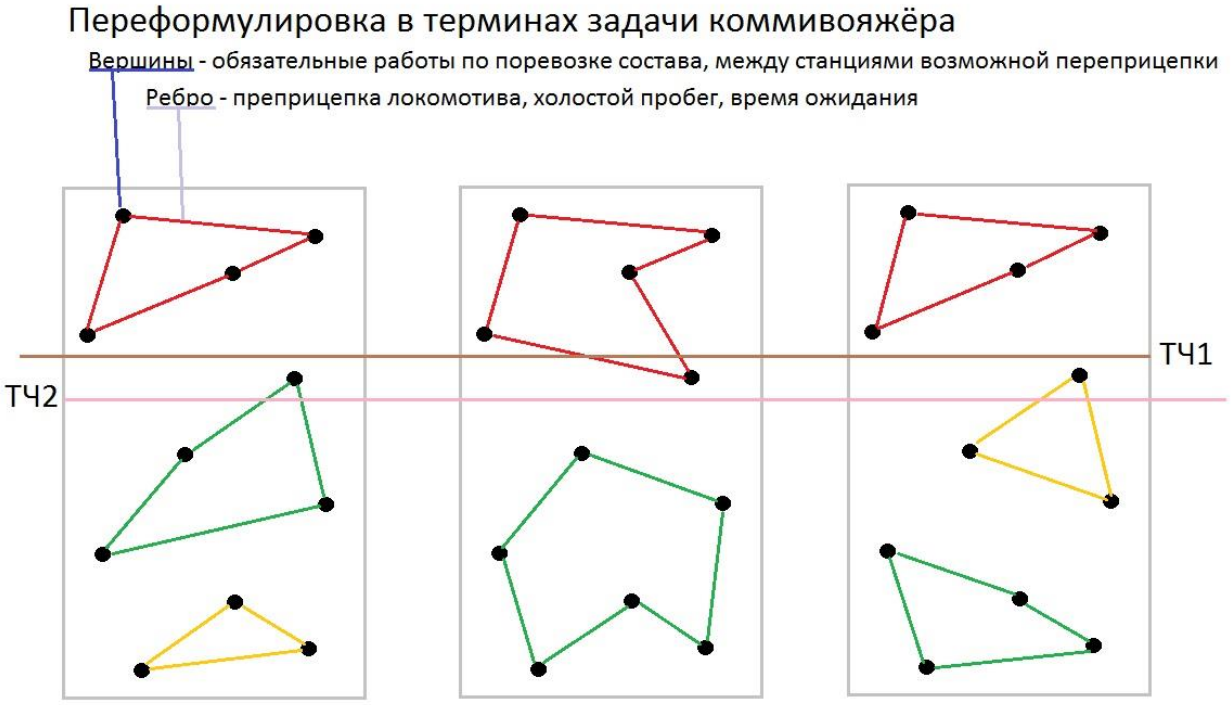
unique schedules takes too long - and many tens of thousands of such checks must be done every day (if they decide on the spot that on a given date it usually takes place without When stopping, the train needs to be stopped and a carriage attached to it, or uncoupled, then this is already a change in the train schedule, and there can be dozens of such changes for each freight train; in addition, due to an accident or other factors, the train may be put on a different route or on a different route; another time). For this purpose, a hashing algorithm was invented - for each unique schedule, which is essentially a text consisting of many sorted lines of a single structure, its hash function was calculated, and the same was calculated for the newly arrived candidate. If the hash function value was new, then the schedule was guaranteed to be new, and if not, then just in case, an exact match of the schedules was checked. This calculation algorithm already worked almost instantly, allowing the identification of a newly arrived schedule among the list of unique ones in microseconds.

As a result, this algorithm was never introduced in the Trans-IT company, because it was then that the company was disbanded by a decision from above, and our department was transferred to VNIIZHT. But I tested the algorithm for normalizing routes through the hashing procedure in an old, still working database on DB2, and then implemented it in the Express-NP system for suburban and long-distance traffic - with its help, quick normalization of the original “raw” data on SQL queries is carried out , without involving complex calculations in other programming languages. In fact, this simple algorithm performs the same database normalization that was previously written in SAS for suburban data, using complex and slow procedures that did not fit into SQL queries and required special processing.

7 Optimization of locomotive operations.

The initial prerequisite for the work was an article in the Gudok newspaper dated February 3, 2014, “Dancing from a diesel locomotive” (http://www.oktmag.ru/archive/?aday=31&amonth=01&ayear=2014, now unavailable on the Internet due to its age). It said that on one section of the Oktyabrskaya Railway (Yanisjärve - Lodeynoye Pole - Volkhovstroy) manual optimization of the movement of locomotives was carried out, as a result of which several locomotives were freed up, and there was a great economic effect. As far as I understand from the article, although non-trivial work was carried out, in general it can be described quite briefly - the search for the optimal route for the operation of several locomotives to achieve a minimum of costs, which can be algorithmized.

When the boss soon called out “We are looking for ideas for development,” I took this article as a basis, reformulated the problem in terms of the well-known “traveling salesman problem,” and programmed its optimization using the ant trail method (the algorithm was taken from an article in Computerra magazine http: //www.computerra.ru/60971/ants/).



What is the crux of the matter? Trains run along the railway and carry goods, but they do not run independently, but are carried by locomotives. One train can travel almost 10,000 km from Moscow to Seoul in a week, while in different places it will be pulled by 5-20 different locomotives, depending on the current situation. And each locomotive has its own locomotive crew, which in most cases,

at the end of the shift, must, together with the locomotive, be at the crew change point at their place of residence, and the locomotive itself, before a new day of work, must undergo maintenance there, and possibly refueling. And at the same time, each locomotive cannot leave the coverage area of its TP (traction part) - for example, an electric locomotive cannot enter a non-electrified area. And between carrying out different work (transporting trains), the locomotive is either in place (idle) or moved to another point to be coupled to another train (idle run). As a result, by organizing the operation of locomotives in different ways, you can at least reduce the cost of work by reducing idle mileage (which is significantly more expensive than idle time), or you can completely get rid of the need for some locomotives by transferring them to hot or cold reserve (this is even cheaper), or transferring to other more popular sections of Russian Railways (doing useful work instead of useless standing).

As a result, I wrote a program in the SAS language, which first learned how to optimize work when the Russian Railways network was randomly divided into different TCs, with a random train schedule, within which it was necessary to optimize the transportation of 100 trains, the routes of which were divided into 1000 segments (analogous traveling salesman problems for 1000 cities). And when the program was fully debugged, and its performance was tested on an even ten times larger volume of data, approval was received from immediate superiors, and an attempt was made to “retrospectively optimize” the work of one specific PM, to perform work actually carried out within one month. The result of the retrospective study was the identification of the possibility of significant savings in effort and money, on the order of hundreds of millions of rubles per year within one PM. At the same time, optimization was carried out only within one TC, without the possibility of redistributing work between two neighboring TCs (the work areas of any two TCs usually have a larger intersection at which work can be carried out by locomotives of any of them), which means a more complete optimization could give even greater effect.

Based on the results of the calculations, in 2018 I made a report at VNIIZhT as part of a project competition, and received second place. First place was taken by the project “introduction of protective tariffs”, which allowed saving on logistics due to the fact that passengers in theory could choose lunch from 3 options, but in reality there was only 1 option for a normal price, and 2 other options for an exorbitant cost, thanks to which Nobody bought them, which means there was no need to purchase them. As a result, my project did not receive any further development.

8 Neural network forecasting.

Since about 2005, my second main task was to build a neural network forecasting system for passenger flows on Russian Railways, for passenger and suburban traffic.

The first versions of the software were written in SAS - the language is very convenient, but slow, and does not support parallel computing. But since it was on it that the automated workplace for processing suburban reporting was written, the program for short-term forecasting of suburban traffic (for a period of 1 month), created at the request of the North-West Industrial Complex, was written on this complex, and is still working.

Since 2015, at the request of the DOSS company in the Trans-IT company, I have been developing forecasting in the more powerful R language. As a result, an array of initial data was accumulated, which made it possible to build a competitive forecasting system that produces the most relevant forecasts. Specifically, thanks to the speed of calculations on multi-core processors, with the ability to parallelize calculations, a program was written for the genetic development of the best neural networks, with the cutting off of the worst options, in which the neural networks competed with each other based on the criterion of the quality of the forecast produced. In this program, each neural network could give a forecast for some part of the source data (for example, for a specific direction or train, or days of the week, or time of day, or season, or class of car, and any combination of these and other factors). And after receiving all the forecasts, for each specific date only the best achievements were taken, and they were transferred to the customer.

As a result, this neural network forecasting complex, working on a personal computer with a 4-core processor, managed to calculate and configure about 10 thousand neural networks per day, and then issue a occupancy forecast for about 500 daily trains for a week in advance, indicating in which direction and how the forecast will change with a small change the cost of selling a ticket in any direction.

During the development of this system, it seems to me that I was able to make an invention in the field of forecasting, namely, I came up with a new principle that allows us to approximate the required parameter with much higher accuracy than conventional methods allow. The bottom line is this: if the predicted random variable has a natural limitation, in my case this is the number of tickets sold, which cannot exceed the number of seats offered for sale, then no attempts to correctly predict it give a good result - the program cannot distinguish where it is dealing with real low demand, and where the culprit is the limitation, and therefore gives a low forecast in all cases. As a result, classical approximation methods never

gave a correlation of the forecast with the original data above 25%, and often even below 20%. The algorithm I developed immediately increased the quality of approximation to 40% and sometimes slightly higher. This only seems a little, but in the case of high-speed trains Moscow-St. Petersburg of the Sapsan type, it made it possible to build a subsequent superstructure of forecasting and analytical proposals for adjusting the composition of trains, for which a retrospective analysis proved significant savings in the use of rolling stock (about 15%), with a slight increase in the level of income from ticket sales.

Since I could not find such algorithms anywhere on the Internet, I consider this algorithm my personal invention, and I will not disclose details without a guarantee of payment. If anyone doubts the existence of a problem, I suggest doing this experiment. As the average statistical expectation of demand on a specific date, take a random variable X, uniform on the interval [0,1], add noise W to it - uniform on the interval [-0.5,0.5], and the signal reflection defect W2 - uniform on [0, 0.1] and take the limiter L-random on the interval [0.5,1.5]. For the input of the neural network, give 2 parameters X and L, and for the output Y=min(k*X+W , L-W2), where the parameter k is considered a priori unknown, but realistically take for example k=1. After this, no matter what internal contents of the neural network you make, a linear combination or a multi-story structure with different functions, a satisfactory prediction will not be achieved quickly.

Unfortunately, due to the collapse of the Trans-IT company, the customer lost interest in the development.

Subsequently, as part of the development of the Express-NP system already at VNIIZhT, there was an attempt to make at least weak forecasting based on SQL programming capabilities, but due to constant delays in putting the Express-NP system into operation (originally planned in 2018, the latest plans were in 2025), I was reoriented to more urgent work, and forecasting passenger turnover at Russian Railways was completely closed.

9 Detection of fraud on the Sakhalin Railway

In 2015, during the construction and research of the capabilities of various predictive models at the proposed test site (Sakhalin Railway), I conducted a full analysis of the primary data on tickets sold, and identified a huge discrepancy between the number of passengers and the amount of revenue. To a first approximation, when passenger traffic doubled on one day compared to others, the amount of revenue increased by only 30-50%, and not doubled.

In the course of considering all possible factors, I discovered that the culprit was the sale of tickets to rented carriages, which were always rented by the same third-party company. As a result, the following theft scheme was revealed.

Since the initial forecast of passenger flows in different seasons could be carried out easily and simply, and thereby roughly predict whether the total number of passengers on the train would be only 3-5 cars, or 8-15 or more, then someone did just that. And on those days when there was a guarantee of a large passenger flow, a third-party company placed an order for attaching additional rented cars to the train, after which it independently sold tickets for them at its own prices, providing its own level of service, different from Russian Railways. But since the cars were rented at 20% of the cost, this company made huge profits from scratch - drawing off effective demand during periods of mass travel, and leaving Russian Railways the right to transport half-empty trains during periods of low load - according to the rules, the train does not have the right be shorter than 8 carriages, and sometimes there are only 3-4 carriages of passengers.

As a result, Russian Railways annually lost about 200-300 million rubles in lost profits on the Sakhalin Railway. After that, I analyzed sales using the same algorithms throughout the Russian Railways network. and revealed annual losses of about 1 billion rubles.

I provided all the findings and evidence obtained to my superiors. Whether they were used or not, I don’t know. But after about a year, the management of this road was replaced in its entirety.

10 Interview at Gazprom for the position of programmer-mathematician

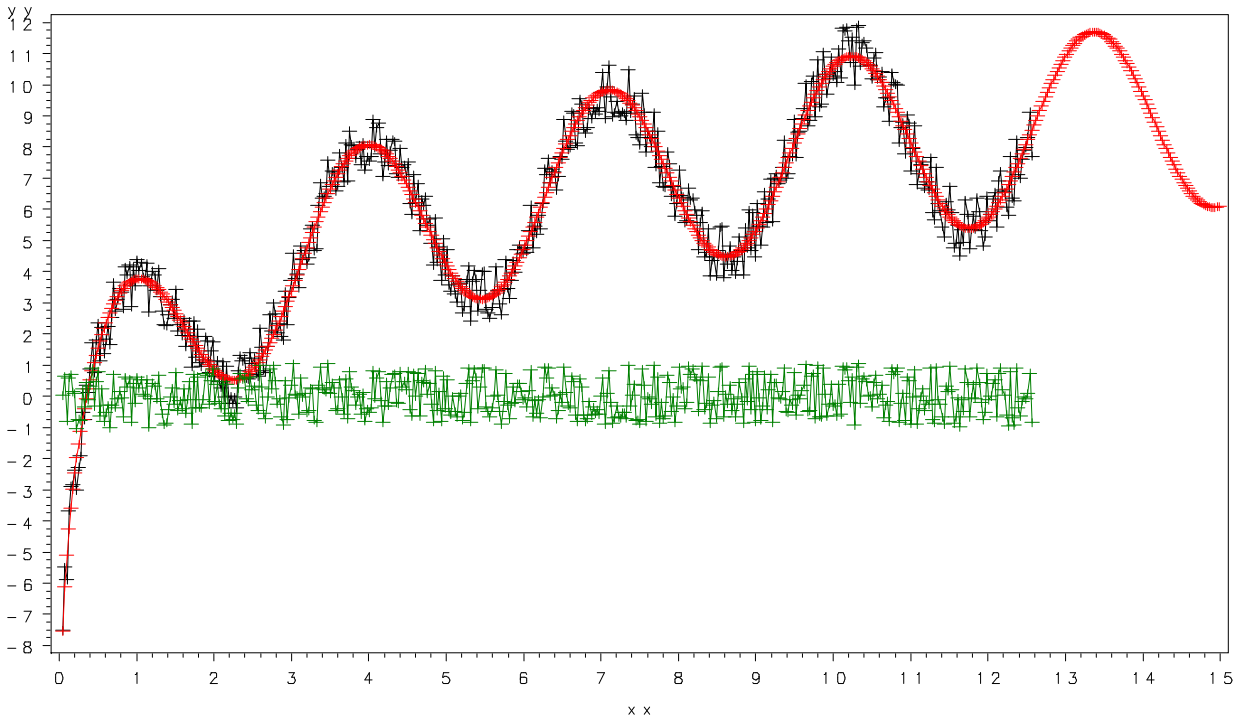
After leaving Russian Railways, in February 2024, representatives of Gazprom contacted me and offered me a vacant position as a programmer-mathematician. To confirm my qualifications, I was given the task of solving 3 problems in 10 days and putting them in the form of presentations. Based on the results of the work done, I was invited to an in-person interview, which I successfully completed, after which I was asked to submit all documents for employment, but my candidacy was blocked by the security department - at that time a criminal case had already been opened against me due to anti-war activities , because of which I had to flee Russia.

The complete archive of job descriptions, task conditions and their solutions consists of files in Russian, I can send it upon request.

Brief description of tasks:

10_1

Given an artificially created time series, as on a graph, it is necessary to predict its behavior for the near future. I couldn't figure out what exact formula it was described by, I just found two very good approximations to it, and showed that over the required future interval they diverge by a very small value - 1.5% of the size of the random deviation. The graph shows: black – the original values, red – the approximating function, green – the difference between them.



10_2

The second task was closer to reality - to explore options for using two UGS (underground gas storage facilities) and find the optimal option in which by the end of the term the daily return would be the maximum possible. At the same time, the possibility of gas release of each storage facility was described by its own formula.

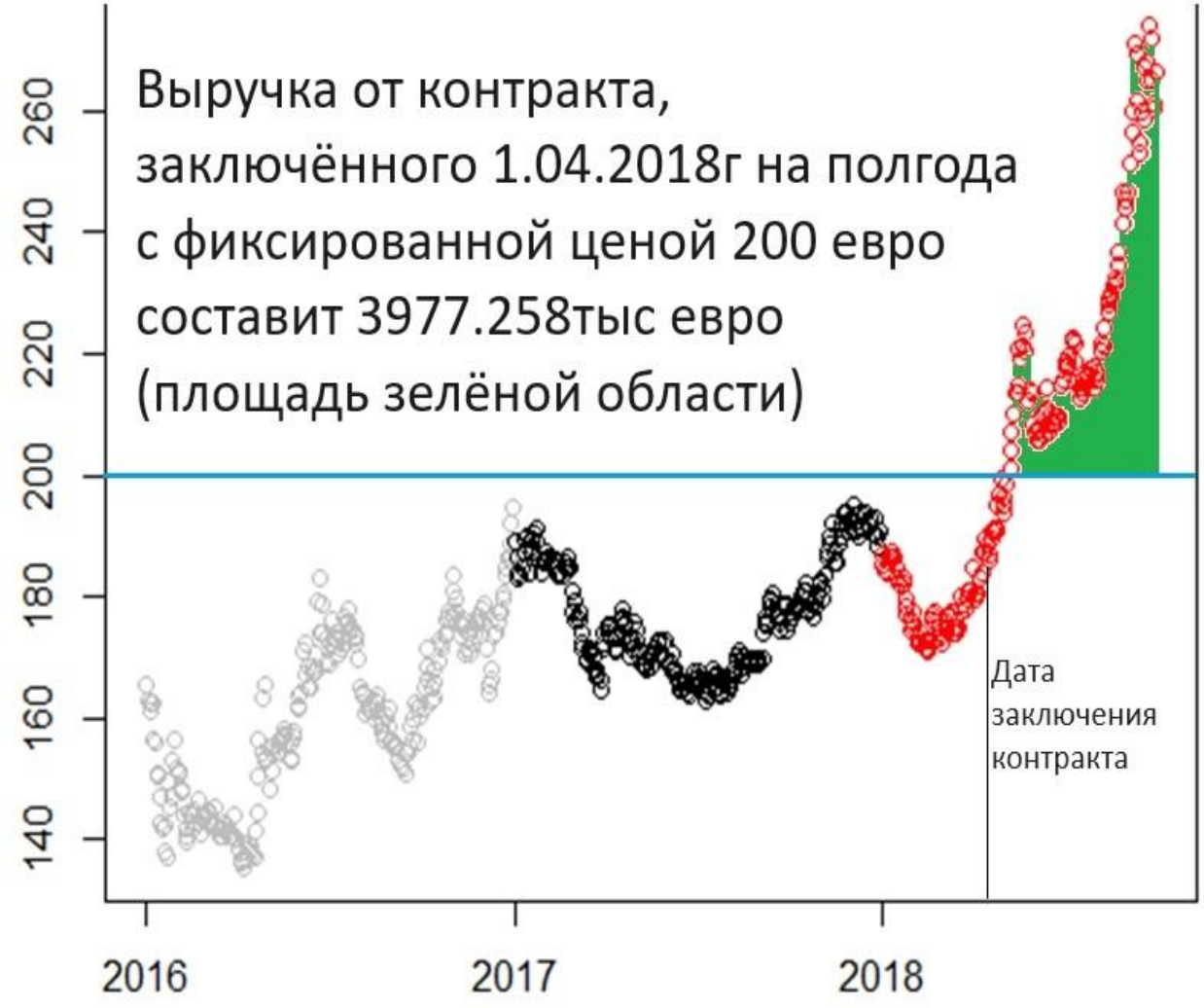
According to the company representative with whom I spoke, I was the first candidate who first made an exact analytical solution to this problem. But since in real life there are many such storages. and each has its own unique gas release function, we had to write a program that solves the problem of optimizing, in the general case, many gas storage facilities with completely different gas release schedules.

10_3

The last task was the most difficult because it was based on real data - according to the gas cost schedule for almost 3 years, it was necessary to calculate the optimal cost of the gas supply option for the next six months, with a fixed supply price. Here I have already managed to give only the interval of the most probable values of the future behavior of the gas price, and, accordingly, the real value of the option. The final answer was as follows: “The average contract revenue under different assumptions is in the range of 5.5-11.2 million euros, and it is impossible to calculate it more accurately within the framework of the available tools.”

The provided graph shows real gas prices for the initially given period, and an example of calculating the cost of a contract if it had been concluded six months

earlier than the required date.



All graphs are taken from final interview presentations.

11 Writing a book on astrophysics

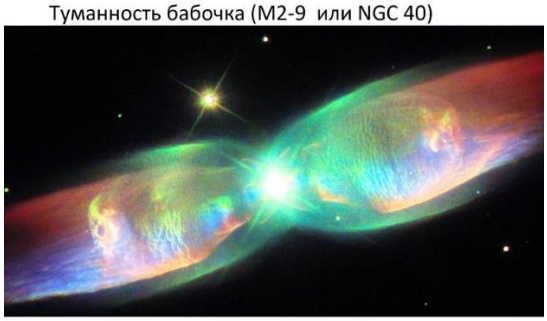
Several years ago, I became interested in issues of astrophysics, I was not satisfied with general words about the process of star formation, and being an expert in mathematics and physics, I created my own theory of the process of formation of dwarf stars. As it turned out later, it is in some ways similar to the theory put forward by the American astrophysicist Carl Rose, but with serious differences.

I examined precisely the process of star formation, and came to the conclusion that in the process of its formation, it simply had to have a central core consisting of heavy elements. Since such a picture of the structure of a star contradicts all modern ideas about it, I began research into the possibility of confirming or refuting my theory. As a result, I found many confirmations of it, and not a single refutation. Along the way, based on this theory, I was able to provide explanations for many phenomena incomprehensible to modern science. I will briefly list the main ones.

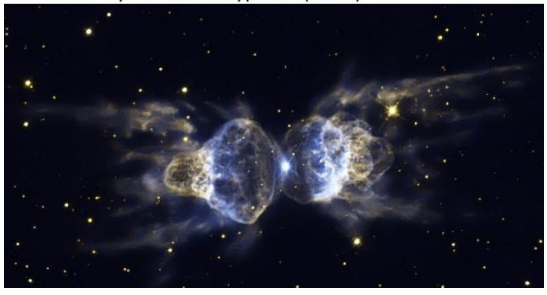
11_1 My theory explains not only the formation of stars, but also the planets around them, and as a result provides an explanation for why the inner planets (Mercury, Venus, Earth and Moon) have a metallic core. But Mars, which has a higher average density than the Moon and a higher iron content on the surface of the planet, is deprived of it. My theory also shows why the total content of different elements on the planets is exactly what we observe it now.

11_2 My theory explains the process of formation of bipolar planetary nebulae, fully describing their internal structure, and the reason for the serious difference in the chemical composition of different parts of the gas envelope. Examples of such nebulae are shown in the illustration.

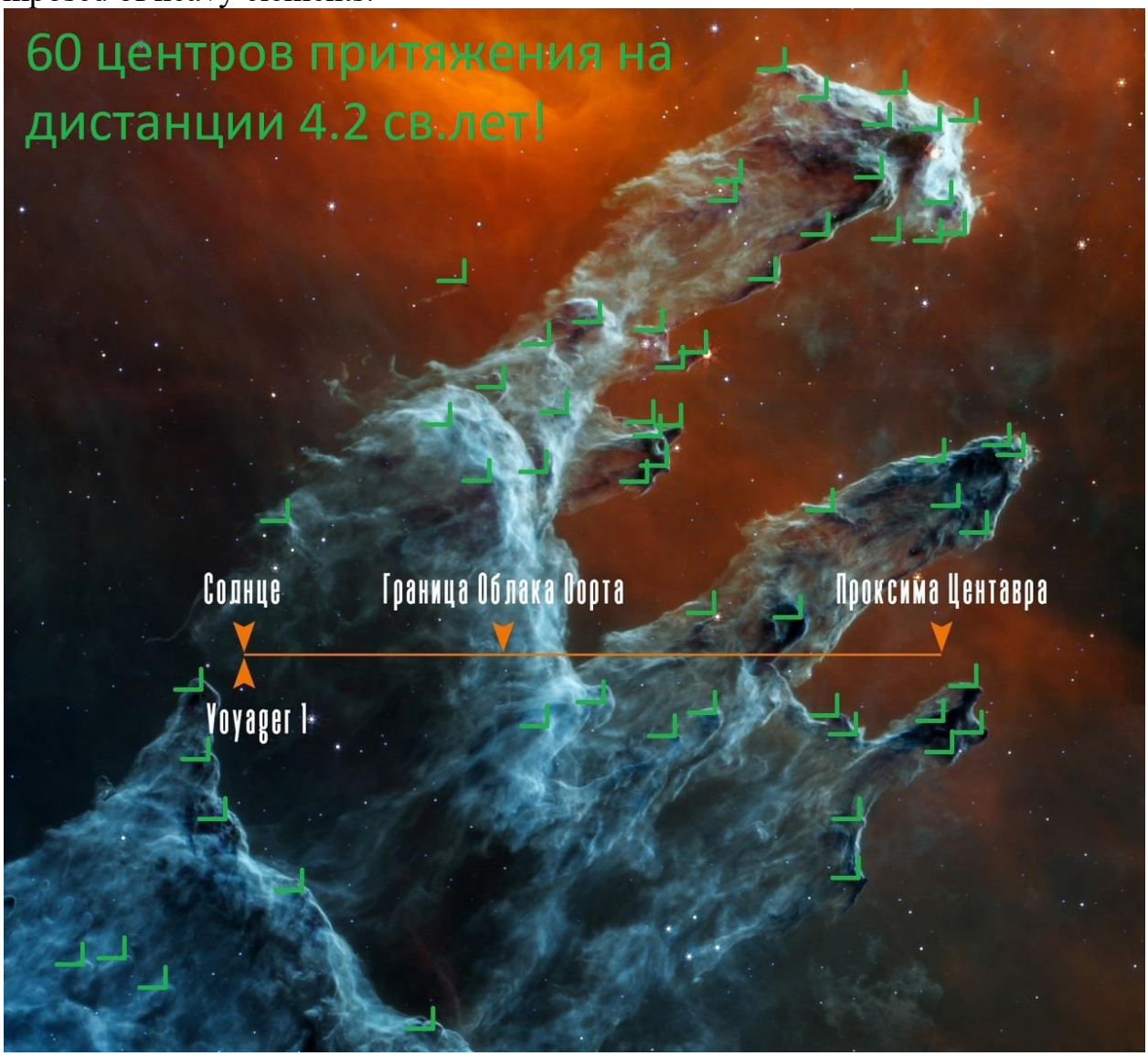
Туманность бабочка (M2-9 или NGC 40)



Туманность Муравей (Mz-3)



11_3 My theory explains the reason for exactly the structure of the Pillars of Creation Nebula that astronomers observe - it was formed not only by the explosion of a nearby supernova, but also by many gravitating protostars composed of heavy elements.

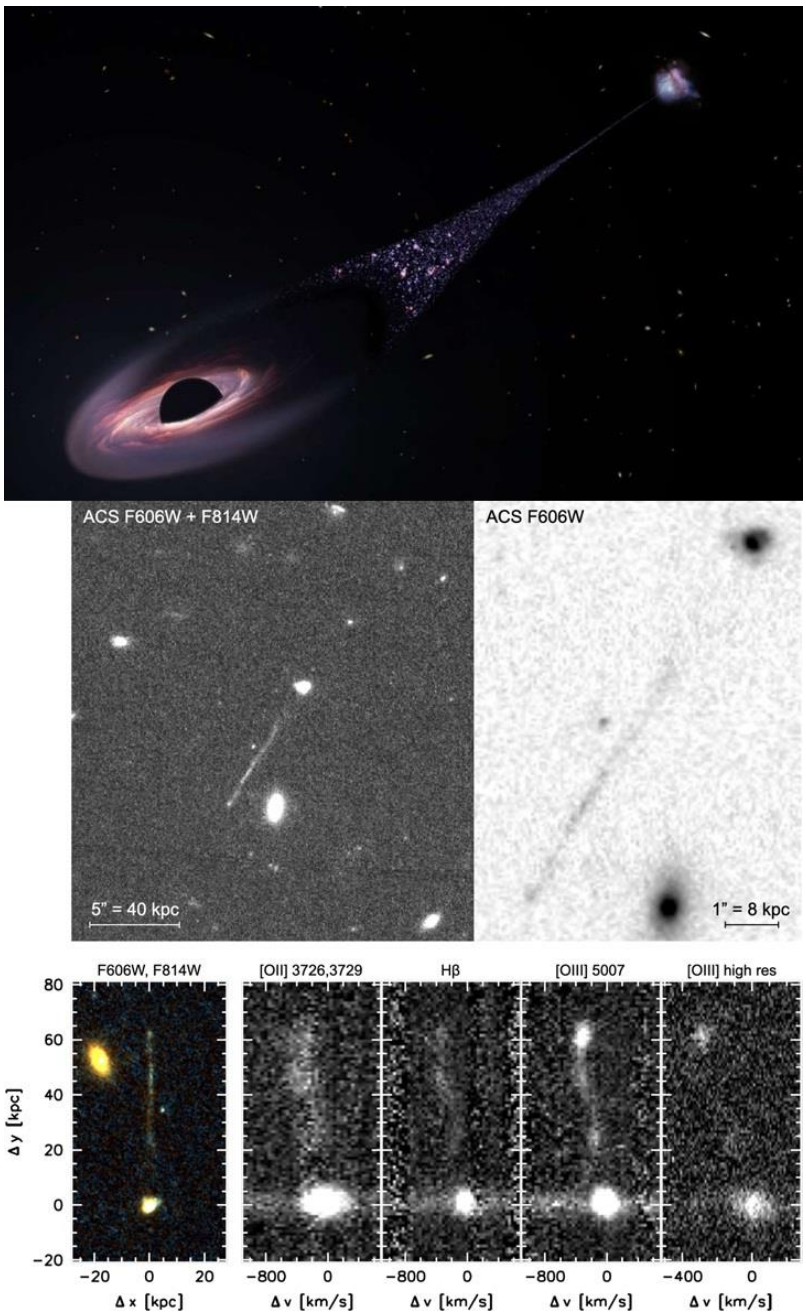


11_4 My theory shows the fallacy of ideas about “super-old stars, almost as old as our universe, and sometimes older than it,” showing in particular that for the luminosity of these stars in the size observed by astronomers, the thermonuclear fuel (hydrogen) contained in their composition would only be enough for a few percent of their lifetime, after which they would be forced to go out - but on the contrary, they shine many times stronger than our sun, having less mass than the sun.

11_5 My theory shows the fallacy of the big bang theory - I purely mathematically proved that this theory contradicts the presence of rotating galaxies. That is, if the Big Bang hypothesis were true, the matter divided into galaxies should have formed non-rotating clumps, which contradicts

observations. My theory, on the contrary, confirms the Big Rebound theory, that is, a cyclical universe. And it shows that in this case, neighboring galaxies should rotate in concert - this was predicted by me before the experimental discovery of this fact in the summer of 2023 - modern astrophysics cannot yet give an explanation for this, but on the contrary, I predicted it and showed the mechanism of such a phenomenon.

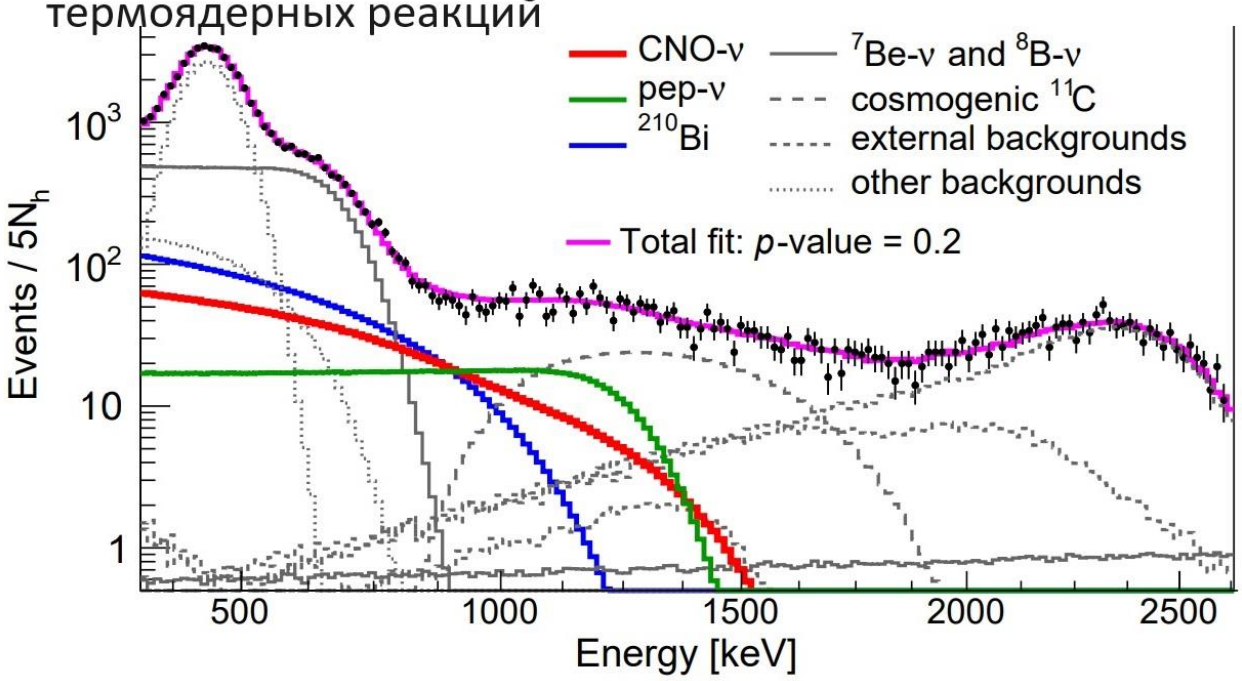
11_6 My theory explains the observation why one black hole, flying out of its galaxy 250 thousand light years away, left behind a luminous trail, shows the mechanism of formation of this glow, and explains its visible internal structure.



11_7 My theory explains why in the photosphere of the Sun, on the one hand, there are absolutely no traces of elements such as lithium or beryllium,

but on the other hand, neutrino astronomy has shown that 8% of all solar neutrinos are generated precisely in the reaction of hydrogen with beryllium - absent in the solar crown of the element. Thus, my theory literally proves the inevitability of the existence of a central core completely devoid of hydrogen.

Спектр энергий всех зарегистрированных нейтрино, в сравнении с несколькими спектрами возможных термоядерных реакций



11_8 My theory also explains the structure of the Pa-30 nebula, which baffles astrophysicists, formed after the collision of two dwarf stars.