

Труды МАИ. 2025. № 143  
Trudy MAI. 2025. No. 143. (In Russ.)

Научная статья  
УДК 519.85  
URL: <https://trudymai.ru/published.php?ID=185656>  
EDN: <https://www.elibrary.ru/WQXZCT>

## ОПТИМИЗАЦИЯ СОСТАВА КОМПЛЕКТАЦИЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

**Алексей Денисович Шаблий**

ПАО «ОАК» «ОКБ Сухого»,

Москва, Россия

[alexey.shabliy@gmail.com](mailto:alexey.shabliy@gmail.com)

**Аннотация:** В настоящей статье рассматривается решение оптимизационной задачи поиска распределения функционала с целью уменьшения стоимости заявленных комплектаций программного обеспечения. Для достижения цели рассматриваемая предметная область описана в виде графа. Описано два решения задачи на графе: математическая формализация с последующей формулировкой задачи линейного программирования и применение генетического алгоритма. Сформулирована и подтверждена экспериментально гипотеза относительно условий применения того или иного метода решения. Сделан вывод о перспективе дальнейшего исследования с целью применения на практике с задействованием реального проекта с открытым исходным кодом полученных теоретических результатов.

**Ключевые слова:** оптимизационная задача, графовая модель, задача линейного программирования, линеаризация, генетический алгоритм

*Для цитирования:* Шаблий А.Д. Оптимизация состава комплектаций программного обеспечения // Труды МАИ. 2025. № 143. URL: <https://trudymai.ru/published.php?ID=185656>

Original article

## **OPTIMIZATION OF THE COMPOSITION OF SOFTWARE PACKAGES**

**Alexey D. Shabliy**

United Aircraft Corporation (Sukhoi Design Bureau),  
Moscow, Russia

[alexey.shabliy@gmail.com](mailto:alexey.shabliy@gmail.com)

**Abstract:** This article solves a topical optimization problem of selecting the composition of software packages. The problem is limited to the subject area of plug-in architectures, and the cost of the formed delivery is chosen as a metric.

First, the subject area is formalized, entities and relationships between them are identified. This formalization is completed by the formation of a graph model and a description of two algorithms: determining the composition of the supplied plug-ins and the cost of their delivery.

In the article, the distribution of functionality among plug-ins is considered as an argument for the operation of the described algorithms and two methods are proposed for finding the optimal distribution: mathematical formalization followed by solving a linear programming problem and the use of a genetic algorithm.

Mathematical expressions for solving an optimization problem are generally nonlinear. However, by applying the restrictions on the values of parameters in expressions described in the article, their linearization is achieved, the implementation of which was carried out in accordance with the formulated and proven theorems.

A hypothesis about the graph configuration is formulated, in which the use of a genetic algorithm will have an advantage over the use of MILP solvers for the formed linear programming problem. This hypothesis is confirmed experimentally: the required running time of MILP solvers shows exponential growth with an increase in the size of the mathematical model, while the required running time of the genetic algorithm does not depend on the size of the model.

To conduct experiments and implement the genetic algorithm, two methods of its operation are considered: with binary data and discrete. The analysis method shows the advantage of working with discrete data over binary data and several configurations of the genetic algorithm with different functions of selection, crossover and mutation are proposed.

In conclusion, the prospects of further research and the feasibility of using open source software solutions to confirm the obtained theoretical results in practice are stated.

**Keywords:** optimization problem, graph model, linear programming problem, linearization, genetic algorithm

**For citation:** Shabliy A.D. Optimization of the composition of software packages. *Trudy MAI*. 2025. No. 143. (In Russ.). URL: <https://trudymai.ru/eng/published.php?ID=185656>

## **Введение.**

В настоящее время решение оптимизационных задач - востребованное направление в сфере информационных технологий, например для программного обеспечения (ПО) [1, 2]. Решение таких задач актуально не только для оптимизации технической составляющей программных решений [3], но и экономической их составляющей. Для технической составляющей актуальными являются вопросы оптимизации работы с разделяемыми ресурсами, организация распределенных вычислений и т.д. Для экономической составляющей важны такие аспекты как оптимизация временных издержек на разработку, сопровождение и послепродажное обслуживание.

Отдельно в экономическом аспекте выделяется стоимость поставляемого решения [4]. Она определяет на каком объеме или сегменте рынка программное решение будет востребовано, а во-вторых, влияет на конкурентоспособность решения на этом объеме рынка.

Для оценивания стоимости существуют различные подходы и на ее конечную величину влияют многие факторы. В настоящей работе в качестве определяющего фактора выделены функциональные требования: их состав и стоимость каждого из них.

Реализация функционального требования обусловлена рядом стоимостных издержек и характеризуется стоимостной величиной, оценка которой индивидуальна, зависит от проекта и устанавливается производителем.

Имея сведения об объеме стоимостных издержек каждого из реализованных требований, можно оценить суммарную их величину применимо для всей версии ПО в целом.

Однако существует возможность поставки одной и той же версии ПО в различных комплектациях [5, 6]. При этом величины стоимостных издержек реализованного требования зависят от реализованных смежных требований и в разных комплектациях могут отличаться друг от друга.

Пример систем, для которых это актуально - плагинные системы [7, 8, 9]. В них интеграционной единицей, несущей конечный объем функционала, является плагин. Так, комплектациями являются различные составы плагинов.

В таких системах выделяют следующие сущности предметной области:

1. функциональные требования;
2. файлы исходного кода;
3. плагины.

Сущности образуют друг с другом следующие связи:

1. в файлах исходного кода на языке программирования описана реализация функциональных требований;
2. файлы исходного имеют зависимости друг на друга и для осуществления поставки все зависимости должны быть разрешены;
3. плагины содержат файлы исходного кода и обеспечивают интеграцию с плагиновой средой выполнения.

Сущности и связи между ними образуют граф [10, 11, 12, 13]. Используя его, можно определить комплектации плагинов. Каждая комплектация реализует конечный и однозначно определяемый состав требований.

Оптимизация состава комплектаций заключается в решении задачи оптимальной декомпозиции файлов исходного кода между плагинами. Критерием оптимальности является суммарная стоимость издержек всех сформированных комплектаций.

### Описание модели

На рисунке 1 приведен пример графа, построенного по описанию сущностей предметной области.

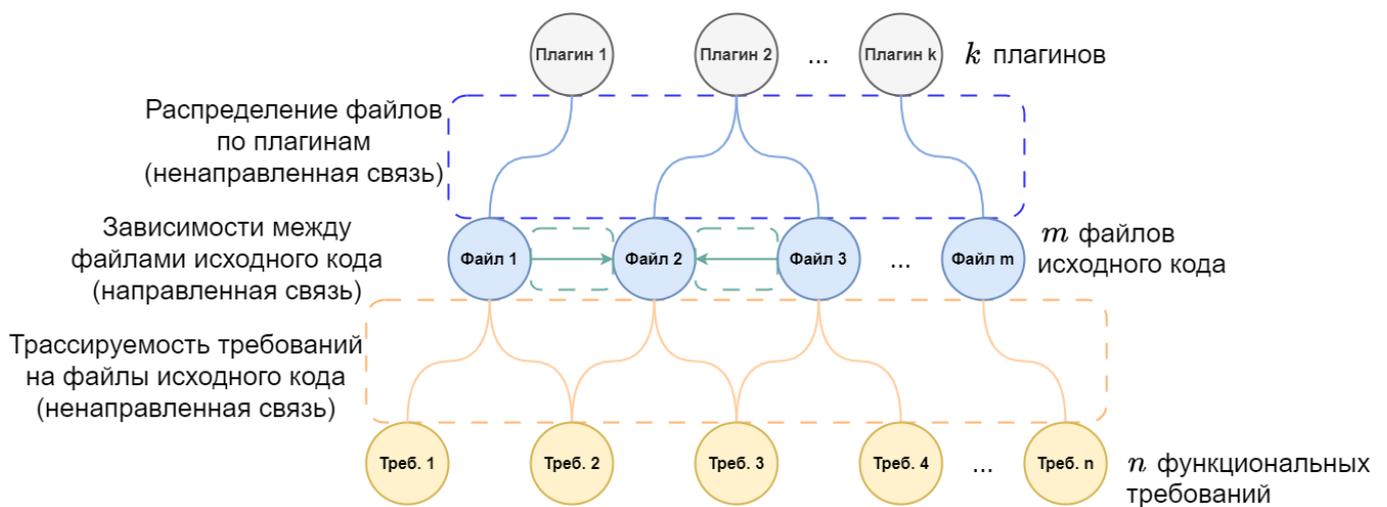


Рис. 1. Графовая модель.

Вершины в графе образуют:

1.  $n$  функциональных требований;
2.  $m$  файлов исходного кода;
3.  $k$  плагинов.

$$n, m, k > 0, \quad n, m, k \in \mathbb{Z} \quad (1)$$

Ребра в графе образуются от:

1. трассируемости требований на файлы исходного кода;
2. наличия функциональных зависимостей между файлами исходного кода;
3. распределения файлов по плагинам.

Полезных требований может быть  $l$  комбинаций. Каждой из таких комбинаций должна соответствовать комплектация, которая обеспечивает выполнение полезных требований. Соответственно должно быть  $l$  комплектаций.

Полезные требования образуют матрицу  $R_{l \times n} = \|r_{i,j}\|$ . В ней номер строки соответствует номеру комбинации. Номер столбца соответствует индексу требования. Если в рамках  $\tilde{l}$ -й комбинации  $\tilde{n}$ -е требование полезно  $r_{i,j} = 1$ . Иначе  $r_{i,j} = 0$ .

$$r_{i,j} \in \{0, 1\}, \quad \sum R_i \geq 1, \quad i = \overline{1, l}, \quad j = \overline{1, n} \quad (2)$$

Трассируемость требований на файлы исходного кода описывает связь требований и файлов. Связь характеризуется степенью ответственности файла для реализации требования.

Эти связи можно представить в виде матрицы  $Q_{n \times m} = \|q_{i,j}\|$ . В ней номер строки - это индекс требования. Номер столбца - индекс файла исходного кода. Элементы  $Q$  описывают долю ответственности файла для реализации требования.

Каждый файл может быть необходим для реализации одного или нескольких требований, а каждое требование реализовано в одном или нескольких файлах.

Допускается, что могут быть файлы, которые не задействованы в реализации хотя бы одного требования, но каждое требование должно быть реализовано. За признак реализованности положена сумма соответствующих элементов матрицы, она должна быть равна 1. Вследствие этого, элементы матрицы  $Q$  удовлетворяют следующим условиям:

$$0 \leq q_{i,j} \leq 1, \quad \sum Q_i = 1, \quad i = \overline{1, n}, \quad j = \overline{1, m} \quad (3)$$

Матрица  $F_{l \times m} = \|f_{i,j}\|$  описывает полезные к поставке файлы исходного кода в каждой из комплектаций. Номер строки соответствует номеру комбинации полезных требований. Номер столбца соответствует индексу файла. Если в рамках  $i$ -й комбинации  $j$ -й файл полезен  $f_{i,j} > 0$ . Иначе  $f_{i,j} = 0$ .

Заметим, что если файл полезен, то нет обязательного условия на равенство значения элемента матрицы. Достаточно чтобы его значение было положительным числом.

$$f_{i,j} \geq 0, \quad i = \overline{1, l}, \quad j = \overline{1, m} \quad (4)$$

Матрицы  $F$  и  $R$  связаны друг с другом через  $Q$ :

$$F \leftarrow R \cdot Q \quad (5)$$

Файлы исходного кода имеют зависимости между друг другом. Зависимость одного файла от другого образует направленную связь, т.е. если первый файл зависит от второго, то второй необязательно зависит от первого.

Такие связи можно представить в виде квадратной матрицы бинарных отношений  $D_{m \times m} = \|d_{i,j}\|$ . В ней номера строк и столбцов соответствуют индексам файлов исходного кода. Если  $i$ -й файл имеет зависимость на  $j$ -й, то  $d_{i,j} = 1$ . Иначе  $d_{i,j} = 0$ . Полагается, что файл не зависит сам от себя, поэтому значения элементов на главной диагонали равны 0.

$$d_{i,j} \in \{0, 1\}, \quad d_{i,j} = 0, \quad i = \overline{1, m}, \quad j = \overline{1, m} \quad (6)$$

Для возможности включения в поставку файла исходного кода необходимо разрешить его зависимости. Разрешение зависимостей подразумевает включение в поставку тех файлов, от которых зависит изначальный. Для дополнительно включаемых файлов так же должны быть разрешены их зависимости.

Зависимости полезных файлов:

$$F' \leftarrow F \cdot D \quad (7)$$

Их зависимости:

$$F'' \leftarrow F' \cdot D = F \cdot D \cdot D = F \cdot D^2 \quad (8)$$

Очевидно, что:

$$F^{(s)} \leftarrow F \cdot D^s \quad (9)$$

Заметим, что максимальная глубина зависимостей не может превышать значение  $m$ , поэтому достаточно разрешить зависимости до этого значения глубины.

Матрица  $\hat{F}_{l \times m} = \|\hat{f}_{i,j}\|$  описывает полезные файлы с учетом разрешения их зависимостей. Аналогично матрице  $F$  в ней номер строки соответствует номеру

комбинации полезных требований, а номер столбца - индексу файла. Для вычисления значений матрицы необходимо просуммировать матрицу полезных файлов и каждую из матриц разрешения зависимостей.

$$\begin{aligned} \hat{F} &\leftarrow F + F' + F'' + \dots + F^m = F + F \cdot D + F \cdot D^2 + \dots + F \cdot D^m = \\ &F \cdot \left( E + \sum_{i=1}^m D^i \right) = \\ &F \cdot \left( D^0 + \sum_{i=1}^m D^i \right) = F \cdot \sum_{i=0}^m D^i \end{aligned} \quad (10)$$

Для значений матрицы  $\hat{F}$  так же, как и для значений матрицы  $F$  характерно, что отсутствует ограничение на положительное значение.

$$\hat{f}_{i,j} \geq 0, \quad i = \overline{1, l}, \quad j = \overline{1, m} \quad (11)$$

В результате вычислений элементы матрицы  $\hat{F}$  могут принимать на столько большие значения, что дальнейшее с ними взаимодействие может быть затруднено. С целью нивелирования данного обстоятельства можно изменить значения матрицы  $\hat{F}$  преобразовав их к бинарным отношениям применив следующую функцию:

$$f_{in}(x) = \begin{cases} 0 & \text{если } x = 0 \\ 1 & \text{если } x > 0 \end{cases} \quad (12)$$

Файлы исходного кода распределены по плагинам. Для включения в поставку файла необходимо включить в комплектацию плагин, который этот файл содержит. Включенный в комплектацию плагин добавляет в поставку все файлы, которые содержит.

Распределение файлов по плагинам образуют матрицу бинарных отношений  $X_{m \times k} = \|x_{i,j}\|$ . В ней номер строки соответствует индексу файла, номер столбца - индексу плагина. Если  $i$ -й файл относится к  $j$ -му плагину, то  $x_{i,j} = 1$ . Иначе  $x_{i,j} = 0$ .

Оптимизационная задача заключается в поиске распределения файлов по плагинам. Плагин может включать несколько файлов, а один файл не может относиться одновременно к разным плагинам. Каждый файл должен относиться к какому-либо плагину. Не каждый плагин должен включать хотя бы один файл. Вследствие этого элементы матрицы  $X$  удовлетворяют следующим условиям:

$$x_{i,j} \in \{0,1\}, \quad \sum X_i = 1, \quad i = \overline{1,m}, \quad j = \overline{1,k} \quad (13)$$

Комплектация плагинов обеспечивает поставку полезного функционала и минимизирует поставку бесполезного. Каждая комплектация включает от 1 до  $k$  плагинов.

Комплектации плагинов для каждой комбинации полезных требований образуют матрицу бинарных отношений  $P_{l \times k} = \|p_{i,j}\|$ . Номер строки соответствует номеру комплектации. Номер столбца соответствует индексу плагина. Если  $j$ -й плагин должен быть включен в  $i$ -ю комплектацию  $p_{i,j} = 1$ . Иначе  $p_{i,j} = 0$ .

$$p_{i,j} \in \{0,1\}, \quad \sum P_i \geq 1, \quad i = \overline{1,l}, \quad j = \overline{1,k} \quad (14)$$

Матрицы  $P$  и  $\hat{F}$  связаны друг с другом через  $X$ :

$$P \leftarrow f_{in}(\hat{F} \cdot X) \quad (15)$$

Алгоритм определения состава комплектаций:

1. лицо, принимающее решение, формирует комбинации полезных требований  $R_{l \times n}$ .

2. для каждой из них определяются полезные файлы исходного кода:

$$F_{l \times m} \leftarrow R \cdot Q \quad (16)$$

3. составляется матрица учитывающая полезные файлы и их разрешенные зависимости:

$$\hat{F}_{l \times m} \leftarrow F \cdot \sum_{i=0}^m D^i \quad (17)$$

4. определяется состав плагинов каждой из комплектаций:

$$P_{l \times k} \leftarrow f_{in}(\hat{F} \cdot X) \quad (18)$$

Поставляемый код — это весь код, который относится к поставляемым в рамках комплектации плагинам. Его описывает матрица бинарных отношений  $\hat{F}_{m \times l} = \|\dot{f}_{i,j}\|$ . В ней номер строки соответствует индексу файла. Номер строки соответствует номеру комплектации.

$$\dot{f}_{i,j} \in \{0,1\}, \quad i = \overline{1,m}, \quad j = \overline{1,l} \quad (19)$$

Матрицы  $\dot{F}$  и  $P$  связаны друг с другом через  $X$ :

$$\dot{F} \leftarrow X \cdot P^T \quad (20)$$

Примечательно, что ограничения значений матриц  $X$  и  $P$  обеспечивают бинарность отношений  $\dot{F}$ . Благодаря этому отсутствует необходимость дополнительной обработки ее значений, например, с использованием функции  $f_{in}$ .

Реализованным называется такое требование, для которого поставлен весь задействованный в реализации код. Состав реализованных требований влияет на стоимость комплектации.

Для определения того, поставляется ли весь код, ответственный за реализацию требования, применяется функция:

$$f_{im}(x) = \begin{cases} 0 & \text{если } x < 1 \\ 1 & \text{если } x \geq 1 \end{cases} \quad (21)$$

Матрица бинарных отношений  $\dot{R}_{l \times n} = \|\dot{r}_{i,j}\|$  описывает состав реализованных требований в каждой из комплектаций. Номер строки соответствует номеру комплектации. Номер столбца соответствует индексу требования.

$$\dot{r}_{i,j} \in \{0, 1\}, \quad i = \overline{1, l}, \quad j = \overline{1, n} \quad (22)$$

Матрицы  $\dot{R}$  и  $\dot{F}$  связаны друг с другом через  $Q$ :

$$\dot{R} \leftarrow f_{im} \left( (Q \cdot \dot{F})^T \right) \quad (23)$$

Стоимость реализованного требования не является постоянной величиной и может быть изменена в зависимости от реализации в комплектации смежных требований.

Величины изменения стоимости реализации требований описывает матрица  $C_{n \times n} = \|c_{i,j}\|$ . В ней номера строк и столбцов соответствуют индексам требований. На ее главной диагонали значениями являются стоимости реализации требования без учета реализации смежных. Прочие элементы в строке указывают на сколько изменится стоимость реализации требования, если будет реализовано соответствующее смежное требование.

$$c_{i,j} \in \mathbb{R}, \quad i = \overline{1, n}, \quad j = \overline{1, n} \quad (24)$$

Заметим, что значением элемента матрицы  $C$  может являться и отрицательное число. Это будет означать, что реализация одного требования уменьшает стоимость реализации другого. Это характерно, например, для дополняющего друг друга функционала.

Стоимость  $i$ -го требования в комплектации  $\tilde{l}$ :

$$\sum_{j=1}^n (c_{i,j} \cdot \dot{r}_{\tilde{l},j}) \quad (25)$$

Данное выражение описывает стоимость требования, как если бы оно входило в комплектацию. Однако даже если само оно не реализовано по этой формуле оно может иметь ненулевую стоимость. Чтобы нивелировать это обстоятельство данное выражение, необходимо помножить на признак реализованности  $i$ -го требования:

$$\dot{r}_{\tilde{l},i} \cdot \sum_{j=1}^n (c_{i,j} \cdot \dot{r}_{\tilde{l},j}) \quad (26)$$

Стоимость одной комплектации - это суммарная стоимость всех реализованных в ней требований:

$$\sum_{i=1}^n \left( \dot{r}_{\tilde{l},i} \cdot \sum_{j=1}^n (c_{i,j} \cdot \dot{r}_{\tilde{l},j}) \right) = \dot{R}_{\tilde{l}} \cdot C \cdot \dot{R}_{\tilde{l}}^T \quad (27)$$

Суммарная стоимость всех реализованных требований во всех комплектациях:

$$\sum_{\tilde{l}} (\dot{R}_{\tilde{l}} \cdot C \cdot \dot{R}_{\tilde{l}}^T) \quad (28)$$

Алгоритм определения стоимости комплектаций:

1. определяется состав комплектаций  $P_{l \times k}$
2. рассчитывается объем поставляемого кода в каждой из них:

$$\dot{F}_{m \times l} \leftarrow X \cdot P^T \quad (29)$$

3. выявляется состав реализованных требований в каждой из комплектаций:

$$\dot{R}_{l \times n} \leftarrow f_{im} \left( (Q \cdot \dot{F})^T \right) \quad (30)$$

4. рассчитать стоимость комплектаций:

$$\sum_{\tilde{l}} (\dot{R}_{\tilde{l}} \cdot C \cdot \dot{R}_{\tilde{l}}^T) \quad (31)$$

Оптимизационная задача может быть решена как задача линейного программирования [12, 14]. С этой целью необходимо построить математическую модель: описать целевую функцию и ограничения на значения задействованных переменных.

Для описания ограничений в общем виде используются следующие переменные:

$$\tilde{l} = \overline{1, l}, \quad \tilde{n} = \overline{1, n}, \quad \tilde{m} = \overline{1, m}, \quad \tilde{k} = \overline{1, k} \quad (32)$$

Например, согласно условиям на значения элементов матрицы  $X$  действуют следующие ограничения:

$$\begin{cases} x_{1,1} + x_{1,2} + \dots + x_{1,k} = 1 \\ x_{2,1} + x_{2,2} + \dots + x_{2,k} = 1 \\ \dots \\ x_{m,1} + x_{m,2} + \dots + x_{m,k} = 1 \end{cases} \quad (33)$$

Используя введенные переменные вышеуказанные ограничения будут записаны так:

$$\sum_{i=1}^k x_{\tilde{m},i} = 1 \quad (34)$$

Формирование целевой функции осуществляется при следовании этапам вышеописанных алгоритмов.

Следуя алгоритму определения состава комплектаций первые три шага не подразумевают наличия переменных, а значит для целевой функции важны значения матрицы  $\hat{F}$ :

$$\hat{F} \leftarrow R \cdot Q \cdot \sum_{i=0}^m D^i = \begin{pmatrix} \hat{f}_{1,1} & \hat{f}_{1,2} & \cdots & \hat{f}_{1,m} \\ \hat{f}_{2,1} & \hat{f}_{2,2} & \cdots & \hat{f}_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{f}_{l,1} & \hat{f}_{l,1} & \cdots & \hat{f}_{l,m} \end{pmatrix} \quad (35)$$

Комплектации плагинов  $P \leftarrow f_{in}(\hat{F} \cdot X)$ :

$$\hat{F} \cdot X = \begin{pmatrix} \sum_{i=1}^m (\hat{f}_{1,i} \cdot x_{i,1}) & \sum_{i=1}^m (\hat{f}_{1,i} \cdot x_{i,2}) & \cdots & \sum_{i=1}^m (\hat{f}_{1,i} \cdot x_{i,k}) \\ \sum_{i=1}^m (\hat{f}_{2,i} \cdot x_{i,1}) & \sum_{i=1}^m (\hat{f}_{2,i} \cdot x_{i,2}) & \cdots & \sum_{i=1}^m (\hat{f}_{2,i} \cdot x_{i,k}) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^m (\hat{f}_{l,i} \cdot x_{i,1}) & \sum_{i=1}^m (\hat{f}_{l,i} \cdot x_{i,2}) & \cdots & \sum_{i=1}^m (\hat{f}_{l,i} \cdot x_{i,k}) \end{pmatrix} \quad (36)$$

Применяя  $f_{in}$  каждому элементу матрицы, необходимо выполнить линейризацию. Для этого используется условно большое число  $M$ , а модель дополняется переменной  $\alpha \in \{0,1\}$ , на которую действуют следующие ограничения:

$$\begin{cases} \alpha < \sum_{i=1}^m (\hat{f}_{l,i} \cdot x_{i,\bar{k}}) + 1 \\ \sum_{i=1}^m (\hat{f}_{l,i} \cdot x_{i,\bar{k}}) \leq M \cdot \alpha \end{cases} \quad (37)$$

Доказательство корректности ограничений методом разбора случаев приведено в таблице 1 с обозначением:

$$\dot{\varepsilon} = \sum_{i=1}^m (\hat{f}_{l,i} \cdot x_{i,\bar{k}}) \quad (38)$$

Таблица 1

Доказательство корректности ограничений от линеаризации  $f_{in}$

$\dot{\varepsilon}$	$\alpha$	$f_1 = (\alpha < \dot{\varepsilon} + 1)$	$f_2 = (\alpha < \dot{\varepsilon} + 1)$	$f_1 \wedge f_2$
0	0	<i>true</i>	<i>true</i>	<i>true</i>
0	1	<i>false</i>	<i>true</i>	<i>false</i>
(0; 1)	0	<i>true</i>	<i>false</i>	<i>false</i>
(0; 1)	1	<i>true</i>	<i>true</i>	<i>true</i>
1	0	<i>true</i>	<i>false</i>	<i>false</i>
1	1	<i>true</i>	<i>true</i>	<i>true</i>
(1; $\infty$ )	0	<i>false</i>	<i>false</i>	<i>false</i>
(1; $\infty$ )	1	<i>true</i>	<i>true</i>	<i>true</i>

На значения рассматриваемых переменных действуют ограничения:

1.  $\dot{\varepsilon}$  не может быть отрицательным;
2.  $\alpha \in \{0,1\}$ .

Следовательно,  $\alpha$  может превышать или быть равной только при  $\dot{\varepsilon} \in [0; 1]$ . При  $\dot{\varepsilon} \in (1; \infty)$  верно всегда, что  $\dot{\varepsilon} > \alpha$ . Поэтому достаточно рассмотрены 4 диапазона значений  $\dot{\varepsilon}$ , которые указаны в таблице 1. В зависимости от диапазона переменная  $\alpha$  принимает значение 0 или 1 согласно линеаризирующей функции. Дизъюнкция

выражений вводимых ограничений показывает корректность значения  $\alpha$  от текущего значения  $\dot{\epsilon}$ .

Корректность вводимых ограничений обоснована истинностью значений  $\alpha$  в каждом из диапазонов  $\dot{\epsilon}$ .

Для будущей программной реализации строгое неравенство должно быть заменено нестрогим. С этой целью вводится постоянная величина  $1/M$  - условно малое число. Тогда, выражение вида  $a < b$  будет записано как  $a + 1/M \leq b$ .

Для линеаризации выражений матрицы  $P$  требуется  $l \cdot k$  дополнительных бинарных переменных  $\alpha$ , которые удобно представить в виде матрицы  $A_{l \times k} = \|\alpha_{i,j}\|$ . Тогда выражение  $f_{in}(\dot{\epsilon})$  заменяется на  $\alpha_{i,\bar{k}}$ , а модель дополняется ограничениями:

$$\begin{cases} \alpha_{i,\bar{k}} + 1/M - \sum_{i=1}^m (\hat{f}_{i,i} \cdot x_{i,\bar{k}}) \leq 0 \\ \sum_{i=1}^m (\hat{f}_{i,i} \cdot x_{i,\bar{k}}) - M \cdot \alpha_{i,\bar{k}} \leq 0 \end{cases} \quad (39)$$

Состав плагинов каждой из комплектаций:

$$P = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,k} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{l,1} & \alpha_{l,2} & \cdots & \alpha_{l,k} \end{pmatrix} \quad (40)$$

Состав поставляемого кода:

$$\dot{F} \leftarrow X \cdot P^T = \begin{pmatrix} \sum_{i=1}^k (x_{1,i} \cdot \alpha_{1,i}) & \sum_{i=1}^k (x_{1,i} \cdot \alpha_{2,i}) & \cdots & \sum_{i=1}^k (x_{1,i} \cdot \alpha_{l,i}) \\ \sum_{i=1}^k (x_{2,i} \cdot \alpha_{1,i}) & \sum_{i=1}^k (x_{2,i} \cdot \alpha_{2,i}) & \cdots & \sum_{i=1}^k (x_{2,i} \cdot \alpha_{l,i}) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^k (x_{m,i} \cdot \alpha_{1,i}) & \sum_{i=1}^k (x_{m,i} \cdot \alpha_{2,i}) & \cdots & \sum_{i=1}^k (x_{m,i} \cdot \alpha_{l,i}) \end{pmatrix} \quad (41)$$

Выражения в матрице  $\dot{F}$  нелинейные, но могут быть линеаризованы благодаря вводу дополнительной бинарной переменной  $\beta$ , потому что переменные  $x$  и  $\alpha$  бинарные. При этом модель дополняется следующими ограничениями:

$$\begin{cases} x + \alpha \leq \beta + 1 \\ \beta \leq x \\ \beta \leq \alpha \end{cases} \quad (42)$$

Доказательство корректности ограничений методом разбора случаев приведено в таблице 2.

Таблица 2

Доказательство корректности ограничений от линеаризации умножения переменных

$x$	$\alpha$	$\beta$	$f_1 = (x + \alpha \leq \beta + 1)$	$f_2 = (\beta \leq x)$	$f_3 = (\beta \leq \alpha)$	$f_1 \wedge f_2 \wedge f_3$
0	0	0	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>
0	0	1	<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>
0	1	0	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>
0	1	1	<i>true</i>	<i>false</i>	<i>true</i>	<i>false</i>
1	0	0	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>
1	0	1	<i>true</i>	<i>true</i>	<i>false</i>	<i>false</i>
1	1	0	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>
1	1	1	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>

Так как  $x, \alpha, \beta \in \{0,1\}$ , можно рассмотреть полный перебор комбинаций значений переменных. В зависимости от комбинации значений  $x$  и  $\alpha$  переменная  $\beta$

принимает значение 0 или 1 согласно линеаризирующей функции. Дизъюнкция выражений вводимых ограничений показывает корректность значения  $\beta$  от текущих значений  $x$  и  $\alpha$ .

Корректность вводимых ограничений обоснована истинностью значений  $\beta$  для каждой из возможных комбинаций значений переменных  $x$  и  $\alpha$ .

Для линеаризации выражений матрицы  $\dot{F}$  требуется  $l \cdot m \cdot k$  дополнительных бинарных переменных  $\beta$ , которые удобно представить в виде матрицы  $B_{l \times m \cdot k} = \|\beta_{i,j}\|$ . Тогда выражение  $x_{\tilde{m},\tilde{k}} \cdot \alpha_{\tilde{l},\tilde{k}}$  заменяется на  $\beta_{k \cdot (\tilde{m}-1) + \tilde{k}, \tilde{l}}$ , а модель дополняется ограничениями:

$$\begin{cases} x_{\tilde{m},\tilde{k}} + \alpha_{\tilde{l},\tilde{k}} - (\beta_{k \cdot (\tilde{m}-1) + \tilde{k}, \tilde{l}} + 1) \leq 0 \\ \beta_{k \cdot (\tilde{m}-1) + \tilde{k}, \tilde{l}} - x_{\tilde{m},\tilde{k}} \leq 0 \\ \beta_{k \cdot (\tilde{m}-1) + \tilde{k}, \tilde{l}} - \alpha_{\tilde{l},\tilde{k}} \leq 0 \end{cases} \quad (43)$$

Состав поставляемого кода с учетом введенных переменных:

$$\dot{F} = \begin{pmatrix} \sum_{i=1}^k \beta_{k \cdot (1-1) + i, 1} & \sum_{i=1}^k \beta_{k \cdot (1-1) + i, 2} & \cdots & \sum_{i=1}^k \beta_{k \cdot (1-1) + i, l} \\ \sum_{i=1}^k \beta_{k \cdot (2-1) + i, 1} & \sum_{i=1}^k \beta_{k \cdot (2-1) + i, 2} & \cdots & \sum_{i=1}^k \beta_{k \cdot (2-1) + i, l} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^k \beta_{k \cdot (m-1) + i, 1} & \sum_{i=1}^k \beta_{k \cdot (m-1) + i, 2} & \cdots & \sum_{i=1}^k \beta_{k \cdot (m-1) + i, l} \end{pmatrix} \quad (44)$$

Определение состава реализованных требований в комплектациях:

$$\dot{R} \leftarrow f_{im} \left( (Q \cdot \dot{F})^T \right) \quad (45)$$

$$(Q \cdot \dot{F})^T = \begin{pmatrix} \sum_{i=1}^m \left( q_{1,i} \cdot \sum_{j=1}^k \beta_{k \cdot (i-1) + j, 1} \right) & \sum_{i=1}^m \left( q_{2,i} \cdot \sum_{j=1}^k \beta_{k \cdot (i-1) + j, 1} \right) & \cdots & \sum_{i=1}^m \left( q_{n,i} \cdot \sum_{j=1}^k \beta_{k \cdot (i-1) + j, 1} \right) \\ \sum_{i=1}^m \left( q_{1,i} \cdot \sum_{j=1}^k \beta_{k \cdot (i-1) + j, 2} \right) & \sum_{i=1}^m \left( q_{2,i} \cdot \sum_{j=1}^k \beta_{k \cdot (i-1) + j, 2} \right) & \cdots & \sum_{i=1}^m \left( q_{n,i} \cdot \sum_{j=1}^k \beta_{k \cdot (i-1) + j, 2} \right) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^m \left( q_{1,i} \cdot \sum_{j=1}^k \beta_{k \cdot (i-1) + j, l} \right) & \sum_{i=1}^m \left( q_{2,i} \cdot \sum_{j=1}^k \beta_{k \cdot (i-1) + j, l} \right) & \cdots & \sum_{i=1}^m \left( q_{n,i} \cdot \sum_{j=1}^k \beta_{k \cdot (i-1) + j, l} \right) \end{pmatrix} \quad (46)$$

Для функции  $f_{im}$  необходимо выполнить линейризацию. Модель дополняется бинарной переменной  $\gamma \in \{0,1\}$ , на которую действуют следующие ограничения:

$$\begin{cases} \sum_{i=1}^m \left( q_{\tilde{n},i} \cdot \sum_{j=1}^k \beta_{k \cdot (i-1) + j, \tilde{l}} \right) \geq \gamma \\ \sum_{i=1}^m \left( q_{\tilde{n},i} \cdot \sum_{j=1}^k \beta_{k \cdot (i-1) + j, \tilde{l}} \right) < M \cdot \gamma + 1 \end{cases} \quad (47)$$

Доказательство корректности ограничений методом разбора случаев приведено в таблице 3 с обозначением:

$$\tilde{\epsilon} = \sum_{i=1}^m \left( q_{\tilde{n},i} \cdot \sum_{j=1}^k \beta_{k \cdot (i-1) + j, \tilde{l}} \right) \quad (48)$$

Таблица 3

Доказательство корректности ограничений от линейризации  $f_{im}$

$\tilde{\epsilon}$	$\gamma$	$f_1 = (\tilde{\epsilon} \geq \gamma)$	$f_2 = (\tilde{\epsilon} < M \cdot \gamma + 1)$	$f_1 \wedge f_2$
0	0	<i>true</i>	<i>true</i>	<i>true</i>
0	1	<i>false</i>	<i>true</i>	<i>false</i>
(0; 1)	0	<i>true</i>	<i>true</i>	<i>true</i>
(0; 1)	1	<i>false</i>	<i>true</i>	<i>false</i>
1	0	<i>true</i>	<i>false</i>	<i>false</i>

1	1	<i>true</i>	<i>true</i>	<i>true</i>
(1; ∞)	0	<i>true</i>	<i>false</i>	<i>false</i>
(1; ∞)	1	<i>true</i>	<i>true</i>	<i>true</i>

Для линеаризации выражений требуется  $l \cdot n$  дополнительных бинарных переменных  $\gamma$ , которые удобно представить в виде матрицы  $\Gamma_{l \times n} = \|\gamma_{i,j}\|$ . Тогда выражение  $f_{im}(\ddot{a})$  заменяется на  $\gamma_{\tilde{l},\tilde{n}}$ , а модель дополняется ограничениями:

$$\begin{cases} \gamma_{\tilde{l},\tilde{n}} - \sum_{i=1}^m \left( q_{\tilde{n},i} \cdot \sum_{j=1}^k \beta_{k \cdot (i-1) + j, \tilde{l}} \right) \leq 0 \\ \sum_{i=1}^m \left( q_{\tilde{n},i} \cdot \sum_{j=1}^k \beta_{k \cdot (i-1) + j, \tilde{l}} \right) + 1/M - (M \cdot \gamma_{\tilde{l},\tilde{n}} + 1) \leq 0 \end{cases} \quad (49)$$

Состав реализованных требований с учетом введенных переменных:

$$\dot{R} = \begin{pmatrix} \gamma_{1,1} & \gamma_{1,2} & \cdots & \gamma_{1,n} \\ \gamma_{2,1} & \gamma_{2,2} & \cdots & \gamma_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{l,1} & \gamma_{l,2} & \cdots & \gamma_{l,n} \end{pmatrix} \quad (50)$$

Стоимость реализованного  $\tilde{n}$ -го требования в  $\tilde{l}$ -й комплектации:

$$\dot{R}_{\tilde{l},\tilde{n}} \cdot C_{\tilde{n}} \cdot \dot{R}_{\tilde{l}}^T = \gamma_{\tilde{l},\tilde{n}} \cdot \sum_{i=1}^n (c_{\tilde{n},i} \cdot \gamma_{\tilde{l},i}) = \sum_{i=1}^n (c_{\tilde{n},i} \cdot \gamma_{\tilde{l},\tilde{n}} \cdot \gamma_{\tilde{l},i}) \quad (51)$$

Выражение  $\gamma_{\tilde{l},\tilde{n}} \cdot \gamma_{\tilde{l},i}$  нелинейно. Однако в силу того, что переменные  $\gamma$  бинарные это выражение может быть приведено к линейному виду благодаря вводу дополнительной бинарной переменной  $\varphi \in \{0,1\}$ . Из ассоциативного свойства  $\gamma_{\tilde{l},\tilde{n}} \cdot \gamma_{\tilde{l},i} = \gamma_{\tilde{l},i} \cdot \gamma_{\tilde{l},\tilde{n}}$  следует, что дополнительная переменная  $\varphi$  может быть переиспользована. В силу этого для линеаризации выражения  $\dot{R} \cdot C \cdot \dot{R}^T$  требуется  $l \cdot$

$n \cdot (n + 1)/2$  дополнительных бинарных переменных  $\varphi$ , которые образуют матрицу  $\Phi_{l \times n \times n}$ :

$$\Phi_{\tilde{l}} = \begin{pmatrix} \varphi_{\tilde{l},1,1} & 0 & \cdots & 0 \\ \varphi_{\tilde{l},2,1} & \varphi_{\tilde{l},2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{\tilde{l},n,1} & \varphi_{\tilde{l},n,2} & \cdots & \varphi_{\tilde{l},n,n} \end{pmatrix} \quad (52)$$

Тогда выражение  $\gamma_{\tilde{l},\tilde{n}} \cdot \gamma_{\tilde{l},i}$  заменяется на  $\varphi_{\tilde{l},\max(\tilde{n},i),\min(\tilde{n},i)}$ , а модель дополняется ограничениями:

$$\begin{cases} \gamma_{\tilde{l},\tilde{n}} + \gamma_{\tilde{l},i} - (\varphi_{\tilde{l},\max(\tilde{n},i),\min(\tilde{n},i)} + 1) \leq 0 \\ \varphi_{\tilde{l},\max(\tilde{n},i),\min(\tilde{n},i)} - \gamma_{\tilde{l},\tilde{n}} \leq 0 \\ \varphi_{\tilde{l},\max(\tilde{n},i),\min(\tilde{n},i)} - \gamma_{\tilde{l},i} \leq 0 \end{cases} \quad (53)$$

Суммарная стоимость реализованных требований в  $\tilde{l}$  комплектации:

$$\sum_{i,j=1}^n C_{i,j} \cdot \varphi_{\tilde{l},\max(\tilde{n},i),\min(\tilde{n},i)} \quad (54)$$

Для всех выражений решения оптимизационной задачи выполнена линеаризация, что позволяет использовать их для формулировки задачи линейного программирования.

Задача линейного программирования:

$$\begin{aligned}
& \min_X \sum_{\tilde{l}} \sum_{i,j=1}^n C_{i,j} \cdot \varphi_{\tilde{l}, \max(\tilde{n}, i), \min(\tilde{n}, i)} \\
s. t. & \quad \gamma_{\tilde{l}, \tilde{n}} + \gamma_{\tilde{l}, i} - (\varphi_{\tilde{l}, \max(\tilde{n}, i), \min(\tilde{n}, i)} + 1) \leq 0 & i = \overline{1, n} \\
& \quad \varphi_{\tilde{l}, \max(\tilde{n}, i), \min(\tilde{n}, i)} - \gamma_{\tilde{l}, \tilde{n}} \leq 0 & i = \overline{1, n} \\
& \quad \varphi_{\tilde{l}, \max(\tilde{n}, i), \min(\tilde{n}, i)} - \gamma_{\tilde{l}, i} \leq 0 & i = \overline{1, n} \\
& \quad \gamma_{\tilde{l}, \tilde{n}} - \sum_{i=1}^m \left( q_{\tilde{n}, i} \cdot \sum_{j=1}^k \beta_{k \cdot (i-1) + j, \tilde{l}} \right) \leq 0 \\
& \quad \sum_{i=1}^m \left( q_{\tilde{n}, i} \cdot \sum_{j=1}^k \beta_{k \cdot (i-1) + j, \tilde{l}} \right) + 1/M - (M \cdot \gamma_{\tilde{l}, \tilde{n}} + 1) \leq 0 \\
& \quad x_{\tilde{m}, \tilde{k}} + \alpha_{\tilde{l}, \tilde{k}} - (\beta_{k \cdot (\tilde{m}-1) + \tilde{k}, \tilde{l}} + 1) \leq 0 \\
& \quad \beta_{k \cdot (\tilde{m}-1) + \tilde{k}, \tilde{l}} - x_{\tilde{m}, \tilde{k}} \leq 0 \\
& \quad \beta_{k \cdot (\tilde{m}-1) + \tilde{k}, \tilde{l}} - \alpha_{\tilde{l}, \tilde{k}} \leq 0 \\
& \quad \alpha_{\tilde{l}, \tilde{k}} + 1/M - \sum_{i=1}^m (\hat{f}_{\tilde{l}, i} \cdot x_{i, \tilde{k}}) \leq 0 \\
& \quad \sum_{i=1}^m (\hat{f}_{\tilde{l}, i} \cdot x_{i, \tilde{k}}) - M \cdot \alpha_{\tilde{l}, \tilde{k}} \leq 0 \\
& \quad \sum_{i=1}^k x_{\tilde{m}, i} = 1 \\
& \quad \tilde{l} = \overline{1, l}, \quad \tilde{n} = \overline{1, n}, \quad \tilde{m} = \overline{1, m}, \quad \tilde{k} = \overline{1, k} \\
w. b. & \quad X_{m \times k} = \|x\| & x \in \{0, 1\} \\
& \quad A_{l \times k} = \|\alpha\| & \alpha \in \{0, 1\} \\
& \quad B_{l \times m \cdot k} = \|\beta\| & \beta \in \{0, 1\} \\
& \quad \Gamma_{l \times n} = \|\gamma\| & \gamma \in \{0, 1\} \\
& \quad \Phi_{l \times n \times n} = \|\varphi\| & \varphi \in \{0, 1\} \\
& \quad M \gg 10
\end{aligned}$$

Размер модели:

1. количество переменных:

$$var\_count = m \cdot k + l \cdot (k \cdot (m + 1) + n \cdot (0.5 \cdot (n + 1) + 1)) \quad (55)$$

2. количество ограничений:

$$con\_count = m + l \cdot (k \cdot (3 \cdot m + 2) + n \cdot (1.5 \cdot (n + 1) + 2)) \quad (56)$$

Альтернативой решения оптимизационной задачи методом ее математической формализации является применение эвристических алгоритмов. В настоящей работе рассматривается применение генетического алгоритма [15, 16, 17, 18].

При реализации генетического алгоритма нет необходимости проводить линейризацию модели, однако необходимо определить функции оценки, селекции, кроссовера и мутации [19].

Ранее была описана функция вычисления суммарной стоимости комплектаций (31). В качестве оценки может быть использовано противоположное значение:

$$fitness = -1 \cdot \sum_i (\dot{R}_i \cdot C \cdot \dot{R}_i^T) \quad (57)$$

Для вычисления этого значения на каждой итерации должно производиться преобразование матрицы  $X$  из генерируемого генетическим алгоритмом решения (подобранных значений генов). Рассматривалось два варианта генерируемого решения:

1.  $\dot{X}_{1 \times m \cdot k} = \|\dot{x}\| - m \cdot k$  бинарных генов;
2.  $\ddot{X}_{1 \times m} = \|\ddot{x}\| - m$  генов со значениями в диапазоне  $[1; k]$ .

Преобразование  $\dot{X}$  в  $X$ :

$$x_{i,j} = \dot{x}_{i \cdot (k-1) + j}, \quad i = \overline{1, m}, \quad j = \overline{1, k} \quad (58)$$

Преобразование  $\ddot{X}$  в  $X$ :

$$X = Z \quad (59)$$

$$j = \ddot{x}_i, \quad x_{i,j} = 1, \quad i = \overline{1, m} \quad (60)$$

Преобразование  $\ddot{X}$  в  $X$  гарантирует выполнение (34). Кроме того,  $|\ddot{X}| < |X|$ , что упрощает поиск решения. Поэтому использован этот способ преобразования для генетического алгоритма в различных конфигурациях (<https://github.com/AlexeyShabliy/csv/blob/main/combinations.csv>).

### Эксперименты и обсуждение

Целью проведения экспериментов было сравнение эффективности применения генетических алгоритмов в решении оптимизационной задачи по сравнению с математической формализацией и последующим решением задачи линейного программирования. Ожидаемым результатом является вывод о корректности сформулированной гипотезы относительно разных методов решения оптимизационной задачи.

Была выдвинута гипотеза, что при увеличении размеров задачи линейного программирования (число ограничений и переменных) требуемое время для работы MILP-решателей растет экспоненциально. Следовательно, экспоненциально снижается эффективность их применения при увеличении коэффициентов  $n$ ,  $m$ ,  $k$  и  $l$ . В то время как для генетического алгоритма эффективность снижается при увеличении не всех коэффициентов, а только  $m$  и  $k$ . Значит в задачах большого размера с малыми значениями коэффициентов  $m$  и  $k$  применение генетического алгоритма должно быть более целесообразно.

В экспериментах были задействованы MILP-решатели GNU Linear Programming Kit (glpk), COIN-OR branch and cut (cbc) [20], а также генетический алгоритм в трех описанных ранее конфигурациях.

Для экспериментов были подготовлены синтезируемые данные [23]. Данные были синтезированы так, чтобы для  $i$ -й комбинации коэффициентов  $m \in [10; 25]$ ,  $n \in [2; 2 \cdot m]$ ,  $l \in [2; 2 \cdot n]$  и  $k \in [2; m]$  выполнялось:

$$\begin{cases} var\_count_i = 100 \cdot i \\ con\_count_i \leq con\_count_{i+1} \end{cases} \quad (61)$$

Если было невозможно подобрать значения, удовлетворяющие описанным условиям, то синтез данных переходил к следующей итерации. Число комбинаций было ограничено значением 200, всего было подобрано 193 комбинации коэффициентов.

На основе подобранных коэффициентов (<https://github.com/AlexeyShabliy/csv/blob/main/constraints.csv>) выполнялась генерация матриц  $C$ ,  $D$ ,  $R$  и  $Q$ .

Каждая комбинация коэффициентов (<https://github.com/AlexeyShabliy/csv/blob/main/combinations.csv>) использовалась для генерации 10 вариантов матриц  $C$ ,  $D$ ,  $R$  и  $Q$ . Таким образом всего было подготовлено 1930 наборов данных.

## Результаты

С целью демонстрации зависимости потребного времени от размера модели учитывалось не потребное время для выполнения каждого эксперимента, а усредненное по каждой комбинации коэффициентов.

На рисунке 2 приведен график зависимости среднего времени выполнения от размера модели для решателя glpk.

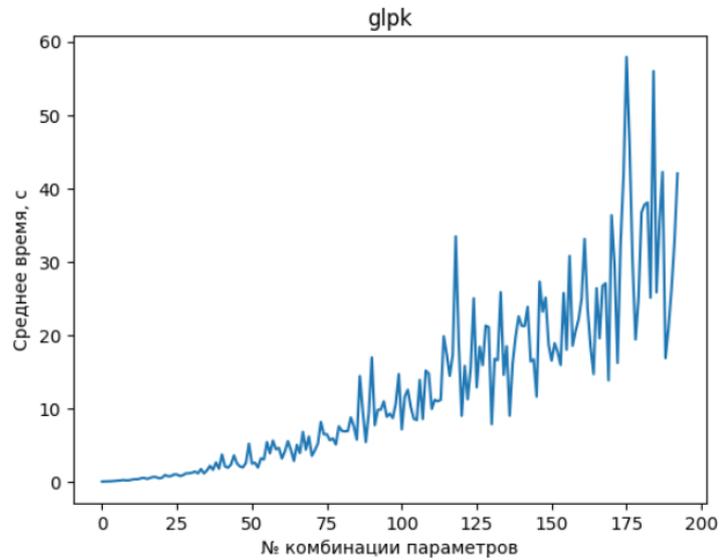


Рис. 2. График зависимости среднего потребного времени решателя glpk

График неравномерный, изменения скачкообразны и для упрощения анализа построен полином второго порядка одной переменной (рисунок 3) [24, 25].

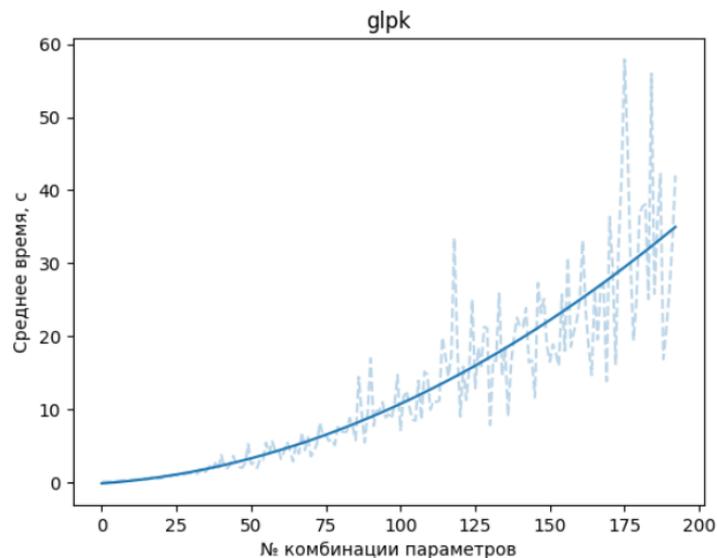


Рис. 3. Аппроксимация данных о среднем потребном времени решателя glpk

Аналогично для решателя cbc (рисунок 4).

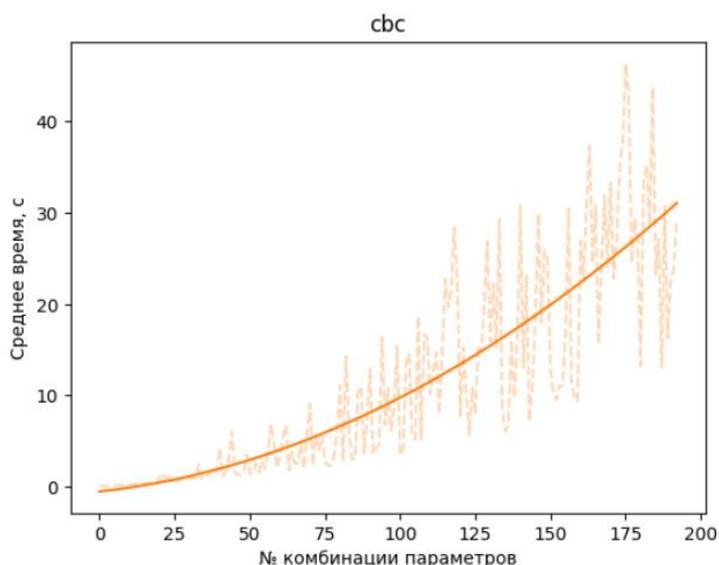


Рис. 4. Аппроксимация данных о среднем потребном времени решателя cbc

Результаты проведения экспериментов подтвердили ожидаемый экспоненциальный рост потребного времени MILP-решателям.

Так же подтверждается гипотеза, что потребное время для решения оптимизационной задачи генетическим алгоритм не зависит от размера модели (рисунок 5).

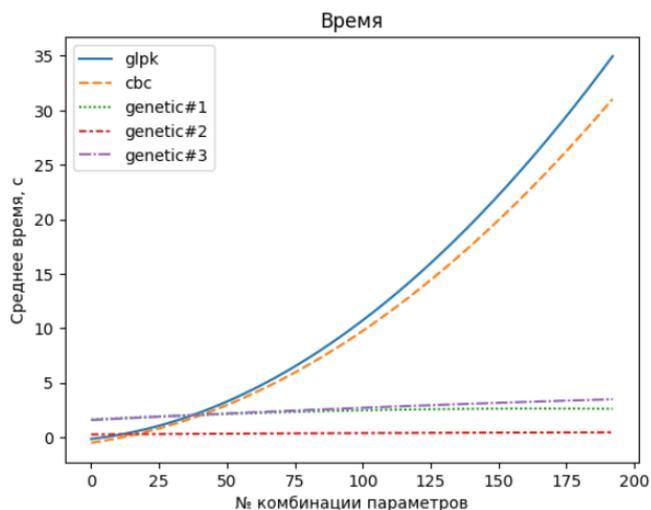


Рис. 5. Сравнение среднего потребного времени

Для вывода о целесообразности применения генетического алгоритма недостаточно опираться на данные о быстродействии решения. Требуется подтверждение объективной оценкой мера оптимизации, которая была достигнута при применении решения. В качестве такой оценки выступает суммарная стоимость комплектаций. Была собрана статистическая информация о том, в скольких случаях предлагалось лучшее решение по сравнению с конкурентами в каждом из 1930 наборов данных. Лучшими считаются одно или несколько решений, которые показали меньшее значение стоимости.

Генетический алгоритм в конфигурации 3 обеспечивал получение наиболее оптимального решения на значительно большем числе наборов данных по сравнению с аналогами (рисунок 6).

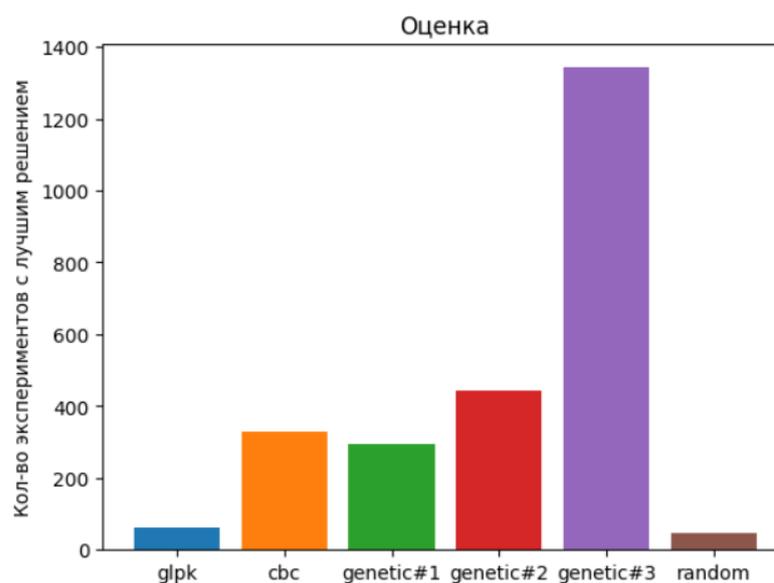


Рис. 6. Сравнение количества достигнутых лучших решений

## **Заключение**

В статье решена оптимизационная задача поиска декомпозиции файлов исходного кода по плагинам по критерию минимальной стоимости формируемой комплектации. Описаны два способа ее решения: математическая формализация с составлением задачи линейного программирования и применение генетического алгоритма. Сформулирована и подтверждена гипотеза об условиях целесообразности применения того или иного решения.

В дальнейших исследованиях предполагается применение описанных механизмов для поиска оптимальной декомпозиции в программных решениях с открытым исходным кодом. Использование их в качестве базиса для формирования исходных данных оптимизационной задачи, ее решения и последующее перераспределение файлов исходного кода по плагинам. Целью такого исследования является получение практических результатов выработанной теории.

## **Список источников**

1. Титов Ю.П., Судаков В.А. Модификация асинхронного метода муравьиных колоний для задач поиска рациональных решений параметрической задачи // Труды МАИ. 2024. № 135. URL: <https://trudymai.ru/published.php?ID=179694>
2. Саяпин О.В., Тиханычев О.В., Безвесильная А.А., Чискидов С.В. Об одной тенденции развития алгоритмов, реализуемых в системах поддержки принятия решений // Программные продукты и системы. 2023. Т. 36, № 3. С. 388–397. DOI: [10.15827/0236-235X.142.388-397](https://doi.org/10.15827/0236-235X.142.388-397)

3. Ай Мин Тайк, Лупин С.А., Федяшин Д.А. Использование библиотеки MPI для параллельной реализации алгоритма полного перебора вариантов // Программные продукты и системы. 2023. Т. 36, № 4. С. 607–614. DOI: [10.15827/0236-235X.142.607-614](https://doi.org/10.15827/0236-235X.142.607-614)
4. Зубкова Т.М., Купырёв С.А. Расчет затрат при проектировании программного обеспечения // Международный научный журнал «Символ науки». 2024. Т. 3, № 4-1. С. 12-18.
5. Гончаров А.Н., Клочай М.С. Обзор модульного проектирования при разработке мобильных приложений: от принципов к реализации // Вестник науки. 2024. Т. 4, № 3 (72). С. 320-327.
6. Земсков М.А. Использование модульных монолитов для разработки масштабируемых web-приложений // Universum: технические науки. 2024. № 10 (127). DOI: [10.32743/UniTech.2024.127.10.18445](https://doi.org/10.32743/UniTech.2024.127.10.18445)
7. Вакушин А.А., Клебанов Б.И. Применение больших языковых моделей в имитационном моделировании // Инженерный вестник Дона. 2024. № 2. С. 624-636.
8. Дагаев Д.В. Инструментальный подход к программированию в системе МультиОберон // Программные системы и вычислительные методы. 2024. № 1. С. 31-47. DOI: [10.7256/2454-0714.2024.1.69437](https://doi.org/10.7256/2454-0714.2024.1.69437)
9. Евглевская Н.В., Казанцев А.А. Обеспечение безопасности сложных систем с интеграцией больших языковых моделей: анализ угроз и методов защиты // Экономика и качество систем связи. 2024. № 4. С. 129-143.

10. Масюков И.И. Метод и устройство расположения задач в реконфигурируемых вычислительных системах // Труды МАИ. 2021. № 120. URL: <https://trudymai.ru/published.php?ID=161427>. DOI: [10.34759/trd-2021-120-13](https://doi.org/10.34759/trd-2021-120-13)
11. Тимофеева О.П., Гордеев М.М., Санников А.Н. Графовые нейронные сети в решении задач с сетевой структурой // Труды НГТУ им. Р.Е. Алексеева. 2023. № 3. С. 43-50. DOI: [10.46960/1816-210X\\_2023\\_3\\_43](https://doi.org/10.46960/1816-210X_2023_3_43)
12. Носков С.И. Задача линейного программирования с альтернативными ограничениями // Известия ТулГУ. 2023. № 7. С. 637-639. DOI: [10.24412/2071-6168-2023-7-639-640](https://doi.org/10.24412/2071-6168-2023-7-639-640)
13. Бородин Д.В., Пруцков А.В. Применение графовой модели и эвристической функции для тестирования программного обеспечения // Известия ТулГУ. 2024. № 3. С. 650-657. DOI: [10.24412/2071-6168-2024-3-650-651](https://doi.org/10.24412/2071-6168-2024-3-650-651)
14. Богданов И.П., Нестеров В.А., Судаков В.А., Сыпало К.И., Топоров Н.Б. Оптимизация загрузки упорядоченной совокупности летательных аппаратов // Известия РАН. Теория и системы управления. 2021. № 3. С. 57–70. DOI: [10.31857/S0002338821030033](https://doi.org/10.31857/S0002338821030033)
15. Кобак В.Г., Белодедов В.А. Влияние миграций при решении неоднородной минимаксной задачи островной моделью // Известия вузов. Северо-Кавказский регион. Технические науки. 2024. № 2. С. 5–10. URL: <http://dx.doi.org/10.17213/1560-3644-2024-2-5-10>
16. Абузьяров А.А., Макаров А.А. Применение и сравнение эволюционных алгоритмов в рамках задачи обучения с подкреплением для неустойчивых систем //

<http://ivdon.ru/ru/magazine/archive/n6y2023/8477>

17. Пантелеев А.В. Метаэвристические алгоритмы оптимизации законов управления динамическими системами. - М.: Факториал, 2020. - 564 с.
18. Пантелеев А.В., Скавинская Д.В. Метаэвристические стратегии и алгоритмы глобальной оптимизации. - М.: Факториал, 2023. - 564 с.
19. Кобак В.Г., Рязанов А.А. Решение задачи коммивояжёра модифицированной моделью Голдберга с использованием поколенческой стратегии при различных эвристиках // Известия вузов. Северо-Кавказский регион. Технические науки. 2024. № 1. С. 5–11. URL: <http://dx.doi.org/10.17213/1560-3644-2024-105-11>
20. Сивакова Т.В., Судаков В.А., Шимко В.С. Исследование методов решения задач смешанного целочисленного линейного программирования // Препринты ИПМ им. М.В.Келдыша. 2024. № 24. 18 с. URL: <https://doi.org/10.20948/prepr-2024-24>  
<https://library.keldysh.ru/preprint.asp?id=2024-24>
21. Алпатов А.Н., Богатырева А.А. Формат хранения данных для аналитических систем на основе метаданных и графов зависимостей между CSV и JSON // Программные системы и вычислительные методы. 2024. № 2. DOI: [10.7256/2454-0714.2024.2.70229](https://doi.org/10.7256/2454-0714.2024.2.70229)
22. Кротов К.В. Алгоритм метода ветвей и границ оптимизации расписаний выполнения пакетов заданий в конвейерных системах // Информационно-управляющие системы. 2023. № 2. С. 15-26. DOI: [10.31799/1684-8853-2023-2-15-26](https://doi.org/10.31799/1684-8853-2023-2-15-26)

23. Авраменко А.Д., Судаков В.А. Создание синтетических графов для задачи коммивояжёра // Препринты ИПМ им. М.В.Келдыша. 2024. № 8. 16 с. URL: <https://doi.org/10.20948/prepr-2024-8> <https://library.keldysh.ru/preprint.asp?id=2024-8>
24. Медведик М.Ю., Барышева А.Д., Демидова А.П., Мекаева В.А., Акашкина Ю.А. Нахождение собственных значений матрицы с помощью метода перемены знака // Вестник Пензенского государственного университета. 2023. № 2. С. 92–98.
25. Торишный Р.О. О применении численных методов второго порядка к задачам стохастического программирования с функцией вероятности // Труды МАИ. 2021. № 121. URL: <https://trudymai.ru/published.php?ID=162664>. DOI: [10.34759/trd-2021-121-17](https://doi.org/10.34759/trd-2021-121-17)

## References

1. Titov Yu.P., Sudakov V.A. Modification of the asynchronous ant colony method for searching for rational solutions to a parametric problem. *Trudy MAI*. 2024. No. 135. (In Russ.). URL: <https://trudymai.ru/eng/published.php?ID=179694>
2. Sayapin O.V., Tikhanychev O.V., Bezvesil'naya A.A., Chiskidov S.V. On one trend in the development of algorithms implemented in decision support systems. *Programmnye produkty i sistemy*. 2023. Vol. 36, No. 3. P. 388–397. (In Russ.). DOI: [10.15827/0236-235X.142.388-397](https://doi.org/10.15827/0236-235X.142.388-397)
3. Ai Min Taik, Lupin S.A., Fedyashin D.A. Using the MPI Library for Parallel Implementation of the Enumeration Algorithm. *Programmnye produkty i sistemy*. 2023. Vol. 36, No. 4. P. 607–614. (In Russ.). DOI: [10.15827/0236-235X.142.607-614](https://doi.org/10.15827/0236-235X.142.607-614)

4. Zubkova T.M., Kupyrev S.A. Calculation of Costs in Software Design. *Mezhdunarodnyi nauchnyi zhurnal «Simvol nauki»*. 2024. Vol. 3, No. 4-1. P. 12-18. (In Russ.)
5. Goncharov A.N., Klochai M.S. Review of Modular Design in Mobile Application Development: From Principles to Implementation. *Vestnik nauki*. 2024. Vol. 4, No. 3 (72). P. 320-327. (In Russ.)
6. Zemskov M.A. Using modular monoliths to develop scalable web applications. *Universum: tekhnicheskie nauki*. 2024. No. 10 (127). (In Russ.). DOI: [10.32743/UniTech.2024.127.10.18445](https://doi.org/10.32743/UniTech.2024.127.10.18445)
7. Vakushin A.A., Klebanov B.I. Application of large language models in simulation modeling. *Inzhenernyi vestnik Dona*. 2024. No. 2. P. 624-636. (In Russ.)
8. Dagaev D.V. Instrumental approach to programming in the MultiOberon system. *Programmnye sistemy i vychislitel'nye metody*. 2024. No. 1. P. 31-47. (In Russ.). DOI: [10.7256/2454-0714.2024.1.69437](https://doi.org/10.7256/2454-0714.2024.1.69437)
9. Evglevskaya N.V., Kazantsev A.A. Ensuring the security of complex systems with the integration of large language models: analysis of threats and protection methods. *Ekonomika i kachestvo sistem svyazi*. 2024. No. 4. P. 129-143. (In Russ.)
10. Masyukov I.I. Method and device for arranging tasks in reconfigurable computing systems. *Trudy MAI*. 2021. No. 120. (In Russ.). URL: <https://trudymai.ru/eng/published.php?ID=161427>. DOI: [10.34759/trd-2021-120-13](https://doi.org/10.34759/trd-2021-120-13)

11. Timofeeva O.P., Gordeev M.M., Sannikov A.N. Graph neural networks in solving problems with network structure. *Trudy NGTU im. R.E. Alekseeva*. 2023. No. 3. P. 43-50. (In Russ.). DOI: [10.46960/1816-210X\\_2023\\_3\\_43](https://doi.org/10.46960/1816-210X_2023_3_43)
12. Noskov S.I. Linear programming problem with alternative constraints. *Izvestiya TulGU*. 2023. No. 7. P. 637-639. (In Russ.). DOI: [10.24412/2071-6168-2023-7-639-640](https://doi.org/10.24412/2071-6168-2023-7-639-640)
13. Borodin D.V., Prutskov A.V. Application of graph model and heuristic function for software testing. *Izvestiya TulGU*. 2024. No. 3. P. 650-657. (In Russ.). DOI: [10.24412/2071-6168-2024-3-650-651](https://doi.org/10.24412/2071-6168-2024-3-650-651)
14. Bogdanov I.P., Nesterov V.A., Sudakov V.A., Sypalo K.I., Toporov N.B. Optimization of the load of an ordered set of aircraft. *Izvestiya RAN. Teoriya i sistemy upravleniya*. 2021. No. 3. P. 57–70. (In Russ.). DOI: [10.31857/S0002338821030033](https://doi.org/10.31857/S0002338821030033)
15. Kobak V.G., Belodedov V.A. Influence of migrations in solving a heterogeneous minimax problem by an island model. *Izvestiya vuzov. Severo-Kavkazskii region. Tekhnicheskie nauki*. 2024. No. 2. P. 5–10. (In Russ.). URL: <http://dx.doi.org/10.17213/1560-3644-2024-2-5-10>
16. Abuzyarov A.A., Makarov A.A. Application and comparison of evolutionary algorithms in the framework of reinforcement learning for unstable systems. *Inzhenernyi vestnik Dona*. 2023. No. 6. (In Russ.). URL: <http://ivdon.ru/ru/magazine/archive/n6y2023/8477>
17. Panteleev A.V. *Metaevristicheskie algoritmy optimizatsii zakonov upravleniya dinamicheskimi sistemami* (Metaheuristic algorithms for optimizing control laws for dynamic systems). Moscow: Faktorial Publ., 2020. 564 p.

18. Pantelev A.V., Skavinskaya D.V. *Metaevristicheskie strategii i algoritmy global'noi optimizatsii* (Metaheuristic strategies and algorithms for global optimization). Moscow: Faktorial Publ., 2023. 564 p.
19. Kobak V.G., Ryazanov A.A. Solution of the traveling salesman problem using a modified Goldberg model using a generational strategy with different heuristics. *Izvestiya vuzov. Severo-Kavkazskii region. Tekhnicheskie nauki*. 2024. No. 1. P. 5–11. (In Russ.). URL: <http://dx.doi.org/10.17213/1560-3644-2024-105-11>
20. Sivakova T.V., Sudakov V.A., Shimko V.S. Study of methods for solving mixed integer linear programming problems. *Preprinty IPM im. M.V.Keldysha*. 2024. No. 24. 18 p. (In Russ.). URL: <https://doi.org/10.20948/prepr-2024-24>  
<https://library.keldysh.ru/preprint.asp?id=2024-24>
21. Alpatov A.N., Bogatyreva A.A. Data storage format for analytical systems based on metadata and dependency graphs between CSV and JSON. *Programmnye sistemy i vychislitel'nye metody*. 2024. No. 2. (In Russ.). DOI: [10.7256/2454-0714.2024.2.70229](https://doi.org/10.7256/2454-0714.2024.2.70229)
22. Krotov K.V. Algorithm of the branch and bound method for optimizing schedules for executing task packages in conveyor systems. *Informatsionno-upravlyayushchie sistemy*. 2023. No. 2. P. 15-26. (In Russ.). DOI: [10.31799/1684-8853-2023-2-15-26](https://doi.org/10.31799/1684-8853-2023-2-15-26)
23. Avramenko A.D., Sudakov V.A. Creating Synthetic Graphs for the Traveling Salesman Problem. *Preprinty IPM im. M.V.Keldysha*. 2024. No. 8. 16 p. (In Russ.). URL: <https://doi.org/10.20948/prepr-2024-8> <https://library.keldysh.ru/preprint.asp?id=2024-8>

24. Medvedik M.Yu., Barysheva A.D., Demidova A.P., Mekaeva V.A., Akashkina Yu.A. Finding the Eigenvalues of a Matrix Using the Sign Change Method. *Vestnik Penzenskogo gosudarstvennogo universiteta*. 2023. No. 2. P. 92–98. (In Russ.)
25. Torishnyi R.O. On the Application of Second-Order Numerical Methods to Stochastic Programming Problems with a Probability Function. *Trudy MAI*. 2021. No. 121. (In Russ.).  
URL: <https://trudymai.ru/eng/published.php?ID=162664>. DOI: [10.34759/trd-2021-121-17](https://doi.org/10.34759/trd-2021-121-17)

Статья поступила в редакцию 20.03.2025

Одобрена после рецензирования 27.03.2025

Принята к публикации 25.08.2025

The article was submitted on 20.03.2025; approved after reviewing on 27.03.2025; accepted for publication on 25.08.2025