

Визуальная аналитика данных в задачах многокритериальной оптимизации

Т.П. Галкин^{1,А}, А.П. Нелюбин^{2,В}, А.А. Галаев^{3,А}, Д.Д. Попов^{4,А},
В.В. Пилюгин^{5,А}, С.Ю. Мисюрин^{6,А,В}

^А Национальный исследовательский ядерный университет «МИФИ», Москва, Россия

^В Институт машиноведения им. А.А. Благонравова РАН, Москва, Россия

¹ ORCID: 0000-0003-2859-6275, z@wqc.me

² ORCID: 0000-0002-7064-3103, nelubin_andrey@inbox.ru

³ ORCID: 0000-0003-3539-3206, aalexgalaev@gmail.com

⁴ ORCID: 0000-0002-3333-749X, DDPopov@mephi.ru

⁵ ORCID: 0000-0001-8648-1690, VVPilyugin@mephi.ru

⁶ ORCID: 0000-0003-1020-0527, symisyurin@mephi.ru

Аннотация

Работа посвящена проблеме анализа многомерных данных в задачах многокритериальной оптимизации. Авторами сформулирована постановка задачи анализа в общем виде и рассмотрен пример проведения такого анализа для задачи оптимизации двухмассовой динамической модели. В качестве метода решения задачи выступает метод визуализации. В работе дано краткое описание метода. Кроме того, подробно описывается конвейер визуализации, применяемый для решения задачи анализа. Сформулированы суждения трёх типов, которые может вынести аналитик. Предложен алгоритм решения задачи анализа с использованием метода визуализации. Описано разработанное программное средство Visual Analytics Tool, реализующее описанный конвейер визуализации и позволяющее производить анализ многомерных данных в соответствии с предложенным алгоритмом. Описаны полученные результаты применения программного средства для анализа данных двухмассовой динамической модели. Проведённый анализ позволяет строить суждения о чувствительности значений целевых функций к изменениям параметров, выделять кластеры решений, наиболее близких к идеалу, а также о влиянии функциональных ограничений на получаемые значения целевых функций.

Ключевые слова: многомерные данные, анализ данных, метод визуализации, многокритериальная оптимизация.

Введение

Современные конструкторские разработки и проектирование представляют собой процесс выбора наиболее подходящей конфигурации разрабатываемого объекта из числа возможных. При этом для сокращения затрат на изготовление прототипов используется моделирование. Выделяется ряд варьируемых параметров, характеризующих прибор, устройство, механизм и т.д., задаются ограничения на значения этих параметров, определяются критерии, по кото-

рым будет производиться отбор моделируемых объектов. Таким образом, перед конструктором возникает задача многокритериальной оптимизации.

Под многокритериальной оптимизацией понимается процесс одновременной оптимизации двух или более конфликтующих целевых функций S в заданной области определения T . Найти наилучшее решение – значит найти одно или несколько таких совокупностей значений переменных P из области определения, при которых значения це-

левых функций были бы приемлемыми для лица, поставившего задачу.

В [1] представлен алгоритм решения оптимизационной задачи с одной целевой функцией. Также существуют различные подходы к решению многокритериальных задач, так, например, в работах [2] и [3] рассматриваются развития американской и французской школ решения многокритериальных задач. Общим в этих подходах является то, что в процессе решения задачи возникают наборы многомерных данных, которые требуется проанализировать. Различаются подходы по тому, как именно проводится анализ этих данных: какие предпосылки делаются и какие суждения получаются в результате анализа.

Во всех вышеупомянутых работах мы видим активное использование визуализации. Программные системы, рассматриваемые в этих работах, позволяют получать изображения, которые помогают аналитику в процессе решения задачи. Эти изображения могут строиться для наглядного представления результатов анализа, а также могут - для представления многомерных данных, возникающих на промежуточных этапах решения, чтобы определить дальнейшие шаги. Таким образом, визуализация естественным образом вошла в концепцию решения задач многокритериальной оптимизации.

Джеймс Томас сформулировал парадигму визуальной аналитики – способ решения задач анализа данных с использованием способствующего интерактивного интерфейса. Она призвана совместить пространственно-образное мышление человека с огромными вычислительными возможностями современных компьютеров. В процессе решения задачи программно может производиться сложное численное моделирование, но при этом обратная связь с аналитиком всегда осуществляется посредством изображений на экране монитора. Конкретной формой визуальной аналитики является метод визуализации, описанный в [4]. Он включает в себя конвейер визуализации и визуальный анализ полученных изображений с по-

следующей интерпретацией результатов анализа. Конвейером визуализации называется интерактивный и итеративный алгоритм построения изображений.

В данной работе описан процесс разработки программы визуального анализа многомерных данных в задачах многокритериальной оптимизации. О преимуществах использования визуализации при решении задач многокритериальной оптимизации рассказывается в работе [5]. В настоящей статье также показаны возможности разработанной программы и примеры её использования. В последствии она станет частью программного комплекса, предназначенного для решения задач многокритериальной оптимизации. Его разработка представляет собой большой проект Института Интеллектуальных Кибернетических Систем НИЯУ МИФИ. Программа визуального анализа данных разрабатывается коллективом УНЛ №227 «Научная визуализация».

1. Формальная постановка задачи анализа

При решении задач многокритериальной оптимизации возникает задача анализа данных. В общем виде она выглядит следующим образом:

Дано:

- $P = \{p_1, p_2, \dots, p_q\}$ – множество действительных переменных, заданных на области определения.
- $C = \{c_1, c_2, \dots, c_m\}$ – m процедурно заданных целевых функций этих переменных.

$$c_z = c_z(p_1, p_2, \dots, p_n), \\ \text{где } z \in [1 .. m].$$

- $G = \{g_1, g_2, \dots, g_k\}$ – k процедурно заданных функций, задающих дополнительные ограничения на область определения T .

$$T = \begin{cases} P_{i-} < p_i < P_{i+} \\ g_j(p_1, p_2, \dots, p_i) \geq G_j, \end{cases}$$

где P_{i-}, P_{i+}, G_j – действительные числа, $i \in [1 .. q], j \in [1 .. k]$.

Требуется:

Сформировать суждения о совместном поведении функций множества C на области их определения.

В статье рассматривается задача многокритериальной оптимизации, описанная в [6], которая сводится к задаче анализа данных с $q = 5$, $k = 3$, $m = 4$. Она имеет вид:

Задана двухмассовая динамическая модель, см. рисунок 1. Здесь M_1 и M_2

массы, K_1 и K_2 – жёсткости, C – демпфирование. Таким образом, $P = \{k_1, k_2, m_1, m_2, c\}$. К массе M_1 приложена гармоническая сила $P \cdot \cos(\omega t)$, $P = 2000$ Н; $\omega = 30$ (c^{-1}).

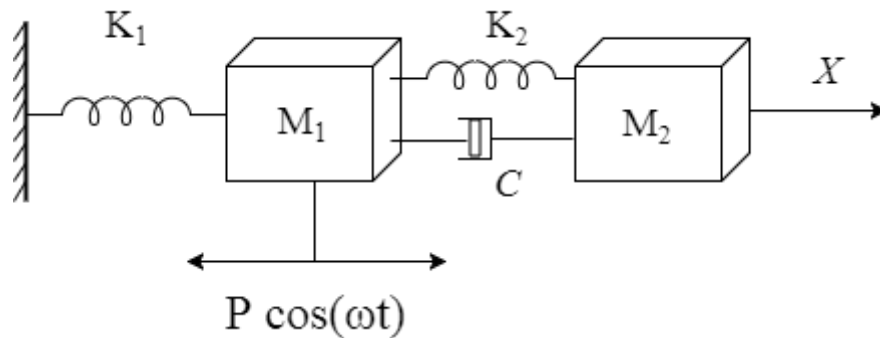


Рисунок 1. Динамическая модель.

Уравнения движения имеют следующий вид:

$$M_1 \ddot{X}_1 + C(\dot{X}_1 - \dot{X}_2) + K_1 X_1 + K_2 (X_1 - X_2) = P \times \cos(\omega t)$$

$$M_2 \ddot{X}_2 + C(\dot{X}_2 - \dot{X}_1) + K_2 X_2 + K_2 (X_2 - X_1) = 0$$

Заданы пять варьируемых параметров:

$$p_1 = K_1, p_2 = K_2, p_3 = M_1, p_4 = M_2, p_5 = C$$

и границы вариации, образующие параллелепипед П,

$$1.1 \cdot 10^6 \leq p_1 \leq 2.0 \cdot 10^6 \text{ (Н/м);}$$

$$4.0 \cdot 10^4 \leq p_2 \leq 5.0 \cdot 10^4 \text{ (Н/м);}$$

$$950 \leq p_3 \leq 1050 \text{ (кг);}$$

$$30 \leq p_4 \leq 70 \text{ (кг);}$$

$$80 \leq p_5 \leq 120 \text{ (Н·с/м).}$$

Имеются три функциональных ограничения Г (на сумму масс и на парциальные частоты):

$$g_1(\alpha) = p_3 + p_4 \leq 1100,0 \text{ (кг);}$$

$$33.0 \leq g_2(p) = \sqrt{\frac{p_1}{p_3}} \leq 42.0 \text{ (} c^{-1} \text{);}$$

$$27.0 \leq g_3(p) = \sqrt{\frac{p_2}{p_4}} \leq 32.0 \text{ (} c^{-1} \text{).}$$

Верхние ограничения на $g_2(p)$ и $g_3(p)$ не являются жесткими и могут в некоторых пределах изменяться.

Путём объединения параллелепипеда П и функциональных ограничений Г получим область определения Т.

Оптимизация проводилась по четырём критериям качества:

$C_1 = X_{1\theta}$ (мм) – амплитуда колебаний первой массы, минимизируется.

$M_1 + M_2$ (кг) – металлоёмкость системы, максимизируется. В настоящей работе этот критерий был заменён на критерий $C_2 = 1120 - (M_1 + M_2)$, чтобы свести задачу к задаче минимизации для всех критериев качества. 1120 кг – максимально возможное значение $M_1 + M_2$, так как в условии задачи присутствуют ограничения $M_1 \leq 1050$ и $M_2 \leq 70$.

$C_3 = X_{1\theta}/X_{1st}$ и $C_4 = \omega/g_2$ (безразмерные величины) – динамические характеристики системы, где X_{1st} – статическое отклонение массы M_1 под воздействием силы P , минимизируются.

В процессе решения задачи с использованием компьютера данная постановка задачи была представлена в виде файла формата lua, в котором в текстовом виде представлены названия переменных, ограничений с соответствующими значениями и целевых функций в

виде функций языка программирования.

Требуется:

Сформировать суждения о совместном поведении критериев качества $C_1 = X_{1\partial}$, $C_2 = -(M_1 + M_2)$, $C_3 = X_{1\partial} / X_{1st}$ и $C_4 = \omega / g_2$ в указанных границах вариации П:

- $1.1 \cdot 10^6 \leq p_1 \leq 2.0 \cdot 10^6$ (Н/м);
- $4.0 \cdot 10^4 \leq p_2 \leq 5.0 \cdot 10^4$ (Н/м);
- $950 \leq p_3 \leq 1050$ (кг);
- $30 \leq p_4 \leq 70$ (кг);
- $80 \leq p_5 \leq 120$ (Н·с/м).

2. Выбор метода решения поставленной задачи

Для решения поставленной задачи анализа многомерных данных был вы-

бран метод визуализации. Суть метода визуализации заключается в том, что исходным анализируемым данным при помощи прикладной программы визуализации ставится в соответствие некоторая их статическая или динамическая графическая интерпретация, которая визуально анализируется, а результаты анализа этой графической интерпретации затем истолковываются по отношению к исходным данным. То есть, метод представляет собой последовательное решение двух задач: задачи непосредственно визуализации и задачи визуального анализа изображений с последующей интерпретацией полученных результатов по отношению к исходным данным.



Рисунок 2. Решение задачи анализа данных методом визуализации.

Алгоритм представления анализируемых данных в виде их некоторого графического изображения предусматривает выполнение следующих шагов:

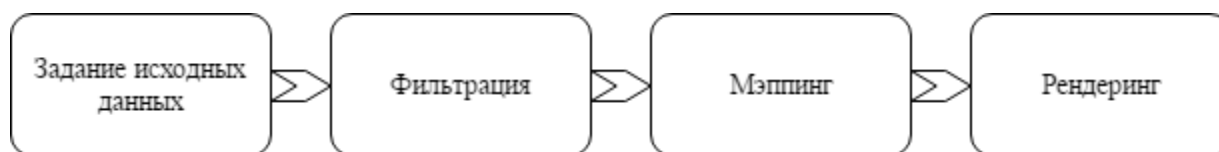


Рисунок 3. Конвейер визуализации.

Совокупность этих шагов, последовательно выполняемых во времени, принято называть конвейером визуализации (visualization pipeline).

2.1. Конвейер визуализации

Задание исходных данных (Sourcing) – процесс получения исходных данных конвейера визуализации. Исходные данные могут быть получены различными способами, как в виде таблицы,

так и в виде значений заданных функций.

Фильтрация (Filtering) - предварительная обработка исходных данных и получение результирующих отфильтрованных данных, которые будут использоваться в дальнейшем для визуализации. Примерами операций этапа фильтрации могут быть интерполяция и сглаживание исходных данных.

Процесс фильтрации не является обязательным и в общем случае может отсутствовать.

Мэппинг (Mapping) – отфильтрованным данным ставится в соответствие описание пространственной сцены, которое включает в себя геометрическую модель этой сцены и описание её оптических характеристик. Геометрическая модель представляет собой совокупность трёхмерных геометрических примитивов или/и сложных геометрических объектов.

Рендеринг (rendering) – получение графического изображения результатов мэппинга. 3D рендеринг имеет ряд параметров, к ним относятся:

1. Камера
2. Источники освещения
3. Среда пространственной сцены

2.2. Визуальный анализ полученного изображения

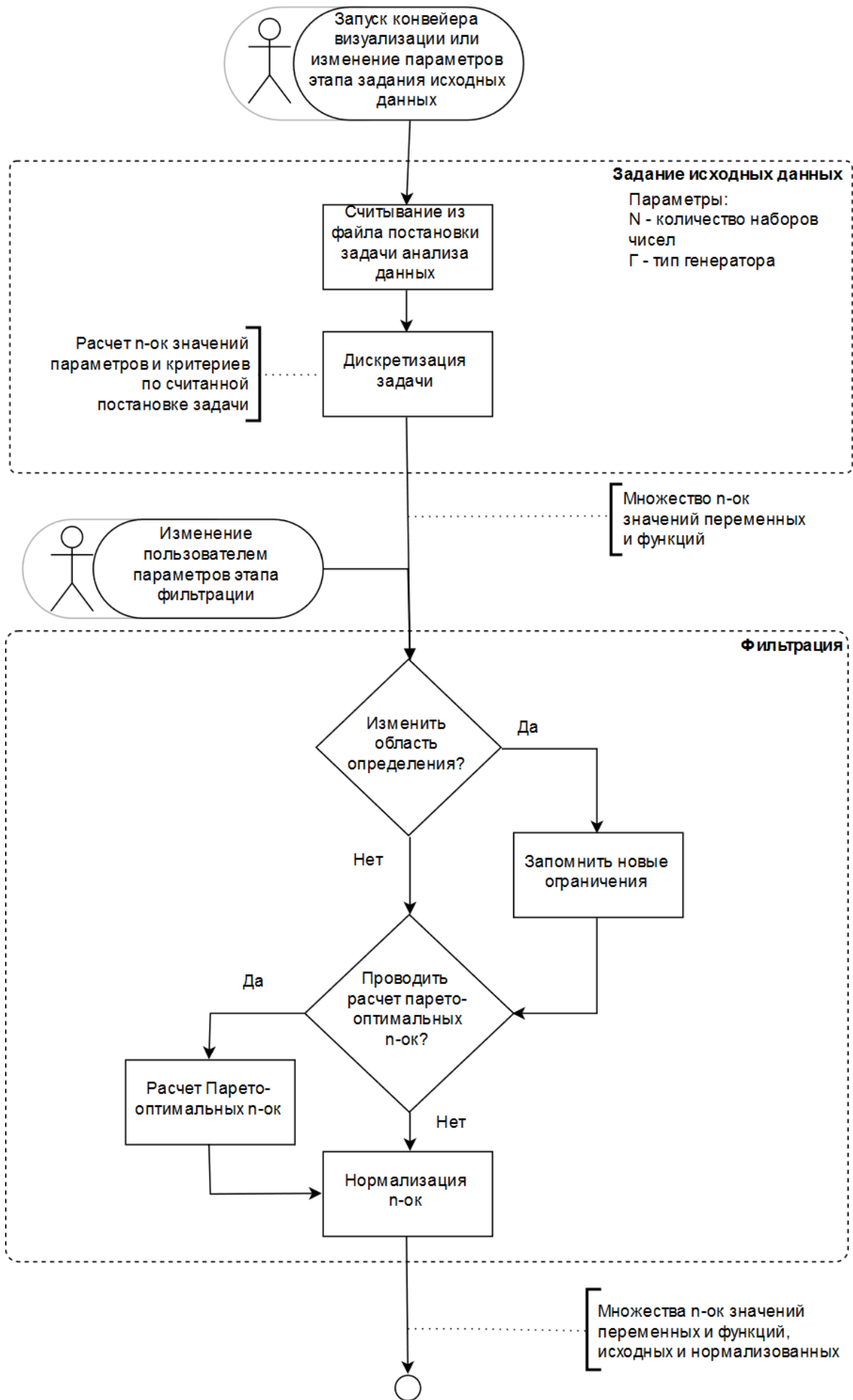
Процесс визуального анализа графического изображения заключается в визуальном анализе той пространственной сцены, которой поставлено в соответствие это изображение. Визуальный анализ пространственной сцены строго не формализуем. В качестве основных задач, решаемых человеком в процессе анализа, можно выделить следующие: анализ формы пространственных объектов, анализ их взаимного расположения и анализ оптических атрибутов пространственных объектов. В случае, когда человек не удовлетворен результатами

анализа, для получения лучших результатов повторно выполняются все или часть шагов конвейера визуализации. В результате, решение задачи анализа исходных данных методом визуализации усложняется и становится итеративным и интерактивным.

3. Разработка алгоритма решения поставленной задачи

Как было указано выше, выбранный метод решения задачи анализа предполагает последовательное решение двух задач: задачи непосредственно визуализации и задачи визуального анализа изображений с последующей интерпретацией полученных результатов по отношению к исходным данным. Рассмотрим разработанный алгоритм конвейера визуализации.

Конвейер состоит из четырёх этапов: Sourcing, Filtering, Mapping и Rendering. Выполнение всех этапов конвейера является интерактивным и итеративным, запускать конвейер визуализации можно повторно с любого этапа. Места, в которых предусмотрено влияние на работу конвейера, помечены пиктограммой человека. Например, получив изображение один раз, пользователь может запустить конвейер не с начала, а с этапа Mapping, предварительно изменив его параметры.



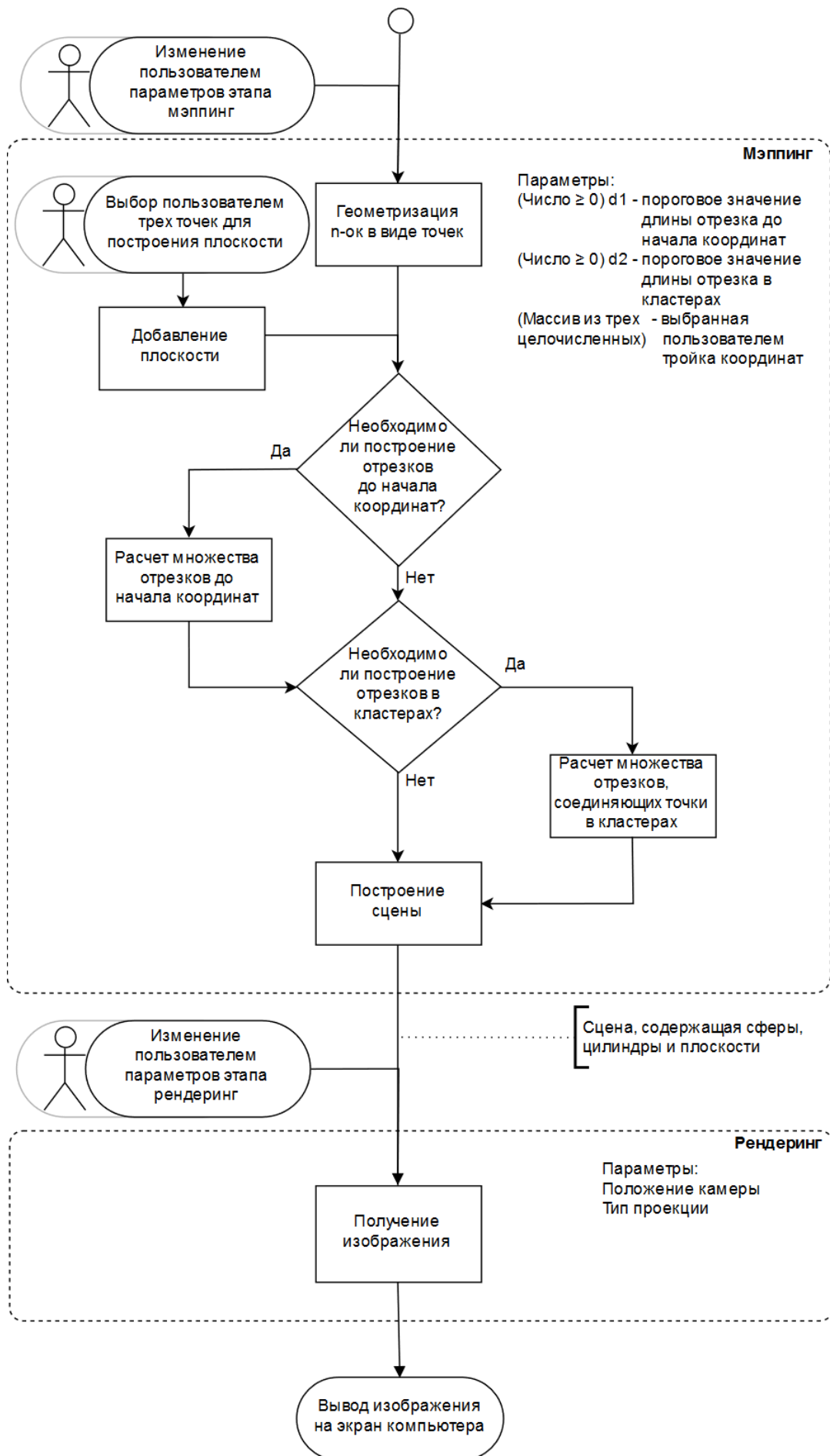


Рисунок 4. Общая схема алгоритма

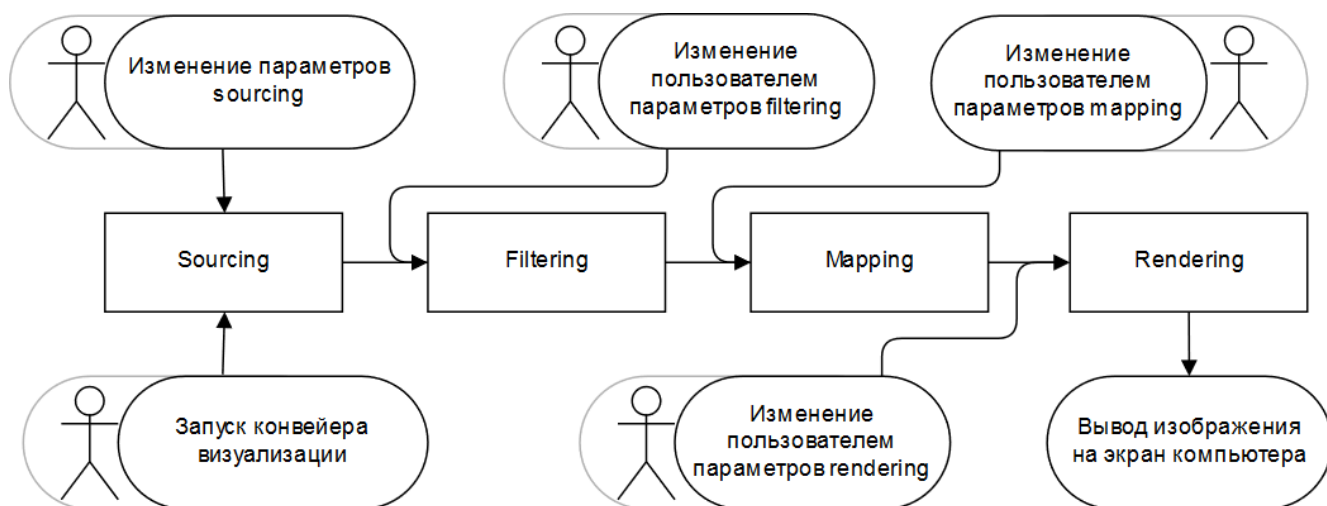


Рисунок 5. Схема конвейера визуализации.

3.1.1. Получение исходных данных

На данном этапе происходит получение исходных данных. Этап является управляемым – имеется возможность интерактивно редактировать входные данные. А именно, пользователю доступно:

- изменение границ параллелепипеда Π границ вариации;
- выбор способа генерации значений параметров – генерация равномерно распределённых случайных величин или генерация квазислучайных последовательностей Соболя;
- выбор количества данных для расчета.

Этап содержит два блока: в первом при считывании из файла постановки задачи анализа данных на выходе имеется её описание в виде процедурно заданных функций; во втором происходит преобразование непрерывных функций в дискретные, иначе говоря, происходит дискретизация задачи. Данный блок позволяет получить упорядоченную совокупность n действительных чисел значений функций и переменных. Такую последовательность далее будем называть n -кой.

Дискретизация задачи

Число генерируемых совокупностей N значений переменных и целевых функций задает пользователь. Таким

образом, каждая n -ка будет иметь порядковый номер $t = \overline{1, N}$. Поскольку каждая n -ка характеризуется упорядоченной совокупностью n действительных чисел, значение переменной или целевой функции будет иметь обозначение p_{it} и c_{zt} соответственно, где $i = \overline{1, q}$ и $z = \overline{1, m}$. Таким образом, мы можем однозначно определять значения рассматриваемых входных переменных и соответствующих целевых функций конкретной n -ки.

Были реализованы два метода генерации значений входных переменных для определения значений целевых функций: случайный и Соболевский. Теоретической базой для реализации алгоритма генерации входных переменных методом Соболя послужила работа [7].

3.1.2. Фильтрация

На данном этапе происходит предварительная обработка данных, полученных на предыдущем этапе: имеется возможность изменять область определения и выделять значения целевых функций, соответствующих Парето-оптимальным вариантам решений оптимизационной задачи. Данные нормализуются, и на выходе имеются множество n -ок значений переменных и целевых функций, а также их нормализованные значения.

Этап состоит из трёх блоков (см. рисунок 4): в первом пользователь может изменить пределы неравенств, задающих функциональные ограничения Γ , определенные в постановке задачи, тем самым меняя область определения; во втором, по желанию пользователя, из множества n -ок значений переменных и функций выделяются n -ки, оптимальные по Парето; третий блок отвечает за нормализацию значений переменных и функций.

Нормализация значений переменных и функций

Для построения геометрической модели полученные данные нормализуются линейно таким образом, чтобы все значения по каждой переменной и функции содержались в пределах отрезка $[0, 100]$.

Вычисляются максимальные и минимальные значения переменных и целевых функций среди всех n -ок. Для каждой упорядоченной совокупности n действительных чисел получим новые множества:

$\tilde{P} = \{\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_q\}$, где значения \tilde{p}_{it} каждой переменной \tilde{p}_i рассчитывается по формуле:

$$\tilde{p}_{it} = \frac{p_{it} - \min_{t=1..N} p_{it}}{\max_{t=1..N} p_{it} - \min_{t=1..N} p_{it}} * 100, i \in [1 \dots q].$$

$\tilde{C} = \{\tilde{c}_1, \tilde{c}_2, \dots, \tilde{c}_z\}$, где значения \tilde{c}_{zt} каждой целевой функции \tilde{c}_z рассчитывается по формуле:

$$\tilde{c}_{zt} = \frac{c_{zt} - \min_{t=1..N} c_{zt}}{\max_{t=1..N} c_{zt} - \min_{t=1..N} c_{zt}} * 100, z \in [1 \dots m].$$

Таким образом, идеальное решение задачи многокритериальной оптимизации, имеющее минимальные значения по каждому из четырех критериев, окажется в начале координат $(0, 0, 0, 0)$ нормализованного пространства критериев.

3.1.3. Мэппинг

На данном этапе происходит геометризация данных и построение простран-

ственной сцены. Упорядоченные совокупности n действительных чисел рассматриваются как точки в n -мерном пространстве. Осуществляется выбор трёхмерного подпространства исходного n -мерного пространства, которое будет использоваться для создания пространственной сцены. Такой подход был предложен в работе [8].

На данном этапе существуют две точки входа: первая – при запуске конвейера визуализации заново с этапа мэппинга или получении управления из этапа фильтрации; вторая – при добавлении пользователем точек для построения плоскости. Сразу после получения управления, происходит геометризация объектов. Множества нормализованных параметров и критериев представляются в виде набора точек в многомерных пространствах \mathbb{P} и \mathbb{C} . В дальнейшем будем называть их пространством параметров и пространством критериев, соответственно.

По желанию пользователя, алгоритм предусматривает добавление в пространства \mathbb{P} и \mathbb{C} множества отрезков между точками и началом координат, отрезков, попарно соединяющих точки, и плоскостей.

Последним блоком этого этапа является построение сцены.

Полученные на этапе фильтрации множества значений с нормализованными данными используются для образования трёхмерного пространства $\{x, y, z\}$ следующим образом: вводится тройка чисел $\{a, b, c\}$, где $a, b, c \in [1 \dots q+m]$, эти числа будут определять выбранное в данный момент подпространство таким образом, что $x = \begin{cases} \tilde{p}_a, & a \leq q \\ \tilde{c}_{a-q}, & a > q \end{cases}$; y и z описываются аналогичным образом.

Затем многомерные точки проецируются на заданное подпространство. В выбранном $\{x, y, z\}$ множество троек (x_i, y_i, z_i) определяется следующим образом: $x_i = \begin{cases} \tilde{p}_{ia}, & a \leq q \\ \tilde{c}_{ia-q}, & a > q \end{cases}$; y_i и z_i задаются аналогичным образом, $i \in [1, N]$.

На множествах $\tilde{\mathbb{P}}$ и $\tilde{\mathbb{C}}$ нормализованных данных, полученных на этапе фильтрации, введём функции:

- $\rho(i, j)$ – функцию расчёта расстояний между точками i и j ,
- $\rho(0, i)$ – функцию расчёта расстояния до начала координат.

Они принимают следующие значения:

- В случае если все элементы $\{x, y, z\}$ содержатся во множестве \mathbb{P} :

$$\rho(i, j) = \sqrt{\sum_{r=1}^q (\tilde{p}_{ir} - \tilde{p}_{jr})^2},$$

$$\rho(0, i) = \sqrt{\sum_{r=1}^q (\tilde{p}_{ir})^2}.$$

- В случае если все элементы $\{x, y, z\}$ содержатся во множестве \mathbb{C} :

$$\rho(i, j) = \sqrt{\sum_{r=1}^q (\tilde{c}_{ir} - \tilde{c}_{jr})^2},$$

$$\rho(0, i) = \sqrt{\sum_{r=1}^q (\tilde{c}_{ir})^2}.$$

- В противном случае, $\rho(i, j)$ не задаётся.

При построении сцены осуществляется добавление набора линий, ограничивающих куб $[0, 100]$, три цилиндра для обозначения направления осей X, Y и Z и текстовые подписи осей. Затем, точки дополняются до трёхмерных сфер, плоскости размещаются на сцене, вписываясь в куб $[0, 100]$, отрезки дополняются до цилиндров.

Под сферой подразумевается пространственный объект со следующими геометрическими и оптическими характеристиками.

Геометрические характеристики:

$$SphG_i = \{p_i(x_i, y_i, z_i), r\},$$

где p_i – координаты центра сферы, r – радиус, задаваемый пользователем, и по умолчанию равный единице.

Оптические характеристики:

$$SphO_i = \{Red_i, Green_i, Blue_i\}$$

$$Red, Green, Blue \in [0; 255]$$

Под цилиндром подразумевается пространственный объект со следующими геометрическими и оптическими характеристиками.

Геометрические характеристики:

$$CylG_i = \{A, B, r\},$$

где A и B – точки, соответствующие основаниям цилиндра,

r – радиус, задаваемый пользователем.

Оптические характеристики:

$$CylO_i = \{Red_i, Green_i, Blue_i\}$$

$$Red, Green, Blue \in [0; 255]$$

$$Red_i = 255 \left(1 - \frac{\rho(A, B)}{d}\right), \quad Green_i = 150 \frac{\rho(A, B)}{d}$$

$$Blue_i = 255 \frac{\rho(A, B)}{d}$$

где $d = d_1$ в случае соединения точек пространств \mathbb{P} и \mathbb{C} отрезками с нулевой точкой,

$d = d_2$ в случае попарного соединения точек пространств \mathbb{P} и \mathbb{C} отрезками.

Под плоскостью подразумевается пространственный объект со следующими геометрическими и оптическими характеристиками.

Геометрические характеристики:

$$PlaneG_i = \{A, B, C\},$$

где A, B и C – точки, заданные пользователем для построения плоскости.

Оптические характеристики:

$$PlaneO_i = \{Red_i, Green_i, Blue_i\}$$

$$Red, Green, Blue \in [0; 255]$$

Результатом выполнения этапа являются геометрическая и оптическая модели сцены. Эти модели используются для получения проекционного графического изображения на этапе rendering.

3.1.4. Рендеринг

Результатом этапа рендеринг является проекционное изображение пространственной сцены. На этом этапе задаются атрибуты визуализации: положение камеры, положение источника света, угол обзора, физические характеристики среды. Пользователь имеет возможность менять положение и направление камеры, выбирать ортографическую или изометрическую проекцию.

3.2. Решение задачи визуального анализа и истолкование результатов по отношению к исходным данным

Результатом выполнения этапа мэппинг являются геометрическая и оптическая модели сцены. Нами были предложены суждения трёх типов, которые можно сделать в процессе визуального анализа полученной пространственной сцены.

1) Суждение о схожести и различии положения сфер (значений целевых функций):

Выделив несколько близко расположенных сфер в одном трехмерном подпространстве цветом и изменив анализируемое трёхмерное подпространство, аналитик может понять, сохраняется ли данная близость в других подпространствах.

2) Суждение о схожести и различии положения сфер по отношению к началу координат:

- Можно делать выводы о близости положения сфер к началу координат в выбранном трёхмерном подпространстве. Меняя подпространства, аналитик может сделать выводы о близости положения сфер к нулю в n -мерном пространстве. В рамках рассматриваемой оптимизационной задачи ноль в n -мерном пространстве является идеальной точкой.
- Меняя пороговое значение d_1 , при котором будут появляться цилиндры между началом координат и сферами, расстояние до которых от начала координат меньше порогового значения, можно сделать выводы о близости значений целевых функций к нулю. Определение степени различия положения сфер друг относительно друга производится, ориентируясь на цвет цилиндров. Синий цвет обозначает расстояние, близкое к пороговому,

красный – малое, по сравнению с ним. Отсутствие цилиндров говорит о большем расстоянии, чем заданное пороговое значение.

3) Суждение о кластерах сфер:

- Меняя пороговое значение d_2 , при котором будут появляться цилиндры между сферами, расстояние между которыми меньше порогового значения, можно будет сделать выводы о близости n -ок друг с другом. Цвет позволит понять, насколько далеко друг от друга они находятся.
- Выделяя цветом интересные сферы, аналитик может, работая в пространстве параметров, изучить поведение сфер, образующих кластеры в пространстве целевых функций.

4. Разработка программы и экспериментальная апробация

Разработка программных модулей интерактивного визуального анализа многомерных данных состояла из четырёх основных частей: реализации интерфейса хранения и обработки полученных многомерных данных, реализации функции получения визуального представления многомерных данных, разработки интерфейса пользователя. Для реализации алгоритмов была написана программа Visual Analytics Tool. Язык программирования – C# с использованием инструмента разработки Unity в модулях расчёта и визуализации данных, а также Lua для описания модели задачи.

Методология разработки предусматривает итеративный подход, который заключается в обсуждении промежуточных результатов на научных семинарах разработчиков. Была использована система контроля версий Gitlab. Она позволяет не только удобно и быстро обмениваться исходным кодом, но и содержит в себе механизм отслеживания

задач, механизм автоматической сборки проекта и сохранения новых (и уже существующих) готовых к запуску версий.

При запуске программы пользователь может ознакомиться с постановкой задачи в общем виде, выбрать файл, содержащий постановку задачи, посмотреть её и перейти к генерации данных. Окно, с которым взаимодействует пользователь, имеет три вкладки, в одной из которых он непосредственно задает параметры дискретизации и принимает решение о выделении сфер, соответствующих Парето-оптимальным вариантам решений оптимизационной задачи. Вторая вкладка позволяет изменить пределы неравенств, задающих функциональные ограничения Г. В третьей пользователь может увидеть сгенерированные числовые значения в табличном виде.

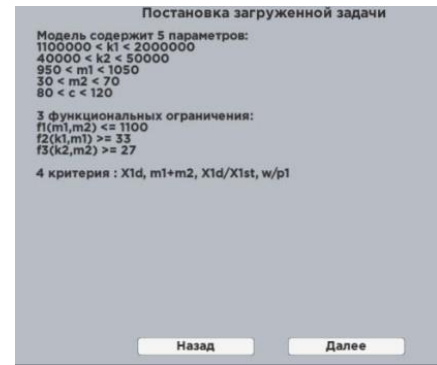
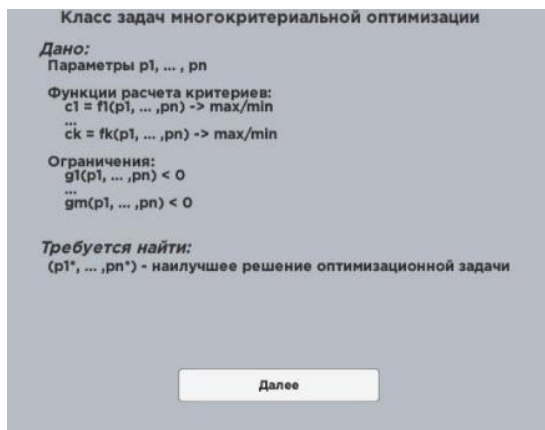


Рисунок 6. Окна программы.

Управление сценой реализовано при помощи нескольких окон, позволяющих изменять значения длин отрезков d_1 и

d_2 , радиус сфер на сцене, выбрать проекцию камеры, включить отображение только отмеченных пользователем сфер,

показать список ограничивающих плоскостей и отредактировать его. Панели можно свернуть.

4.1. Результаты работы программы

Программа позволяет сгенерировать указанное пользователем количество данных, после чего визуально их проанализировать. Программа своевременно реагирует на нажатия и движения мыши. В процессе тестирования генерировались данные о 100'000 точек. При проведении испытаний на различных компьютерах, время расчета и добавления точек на сцену не превышало 60 секунд.

Стоит заметить, что проделанная над пользовательским интерфейсом работа достаточна, чтобы обеспечить удобную манипуляцию многомерными данными для аналитика. Ему предоставлены не только возможности менять подпространство $\{x, y, z\}$ и строить цилиндры между сферами, но также возможность вращения камеры на сцене и переключения её проекции между перспективной и ортогональной. Важно отметить, что пользователю доступна возможность изучить данные в табличном виде.

5. Проведенный анализ

Проведём визуальный анализ сцены, описанный в п. 3.2.

Выбираем облако сфер в пространстве параметров:

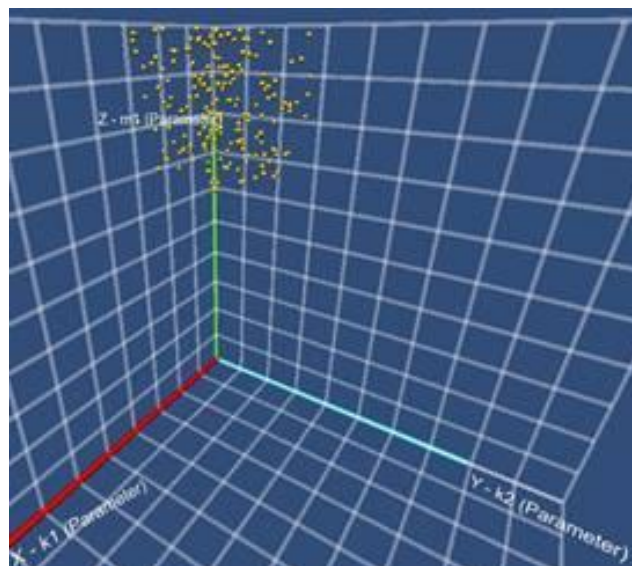
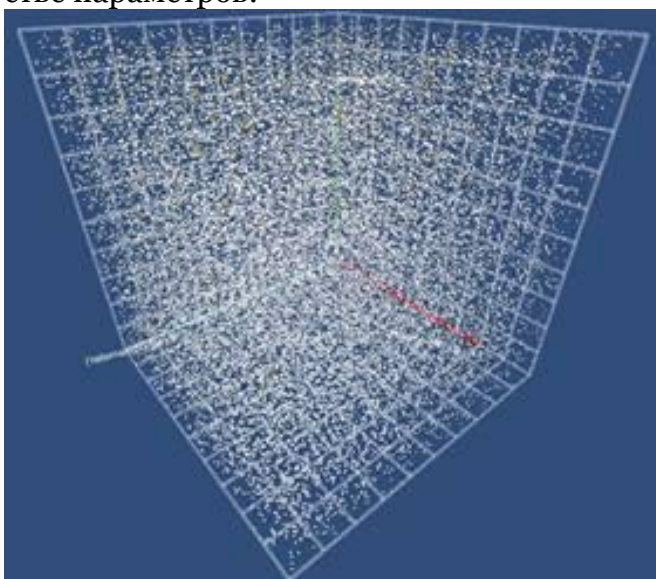


Рисунок 7. Облако сфер

Рассмотрим следующий диапазон значений:

$$\begin{aligned} 1.1 \cdot 10^6 < K_1 < 1.37 \cdot 10^6; \\ 4.0 \cdot 10^4 < K_2 < 4.3 \cdot 10^4; \\ 1020 < M_1 < 1050; \\ 30 < M_2 < 58; \\ 80 < C < 120. \end{aligned}$$

Осуществим переход в пространство критериев и зададим пороговое значение, при котором сферы из облака связываются цилиндрами. Иными словами, образуют кластеры.

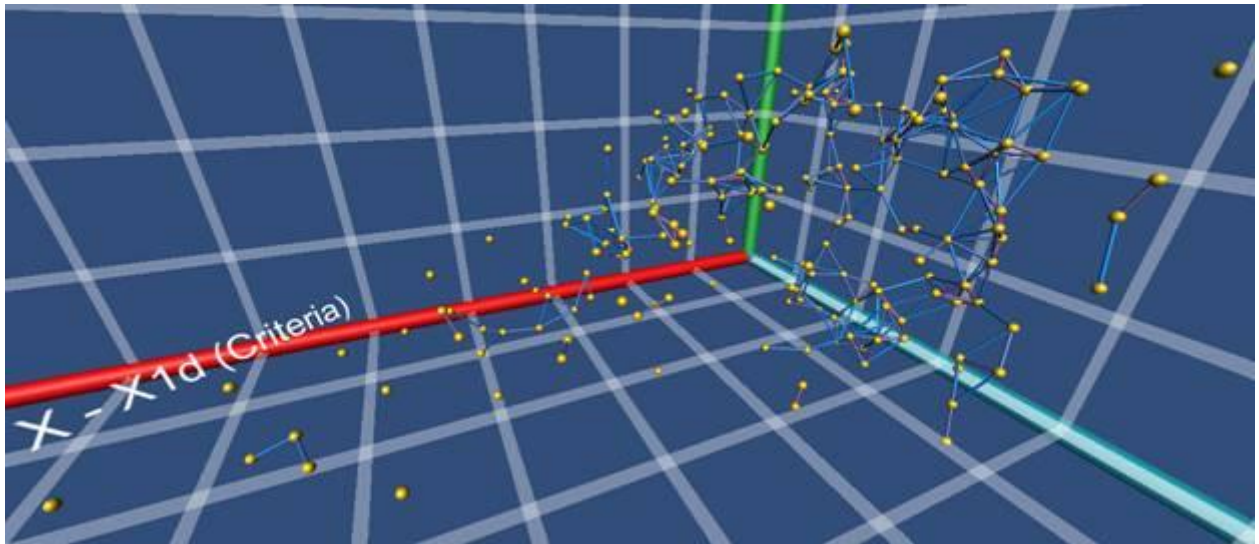


Рисунок 8. Кластеры в пространстве критериев

Имея подпространство, образованное целевыми функциями $X_{1d}, C_2, \frac{\omega}{p_1}$, соответственно по осям абсцисс, ординат и аппликата, изменим подпространство, произведя замену целевой функции $\frac{\omega}{p_1}$ на $\frac{X_{1d}}{X_{1st}}$.

вывод о том, что значения не принятых в рассмотрение критериев различны.

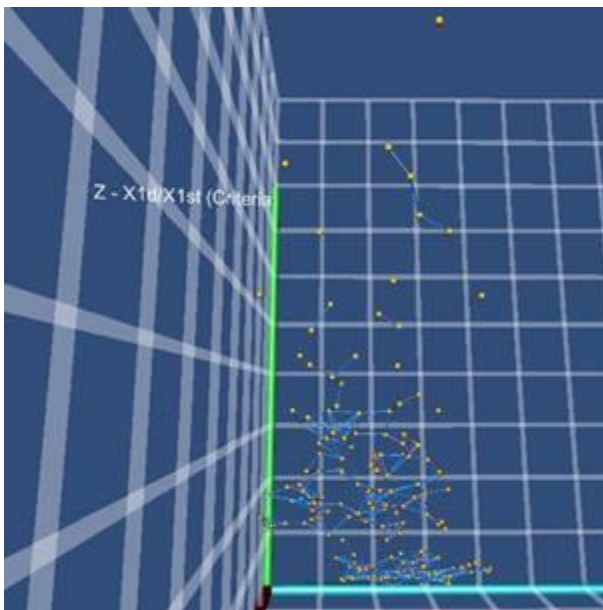


Рисунок 9. Кластеры в измененном подпространстве.

Из рисунка 10 вынесем суждение типа 3 (о кластерах сфер). На рисунке можно заметить, что две пары сфер, визуально находящиеся на одном расстоянии в трехмерном пространстве, имеют попарно соединяющие их цилиндры разного цвета. Из этого можно сделать

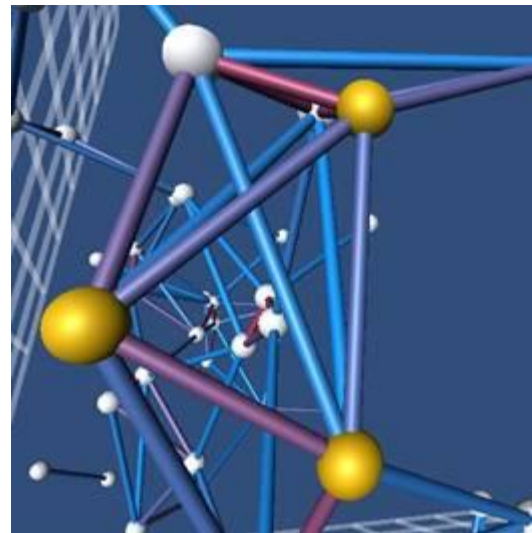


Рисунок 10. Попарное сравнение расстояния между точками.

Также можно заметить, что облако в пространстве параметров образует кластер близко расположенных друг к другу точек, в то время как в пространстве критериев точки более отдалены друг от друга. Таким образом можно сделать вывод, что незначительное изменение в пространстве параметров ведёт к значительному изменению в пространстве критериев. Это наблюдение полезно в контексте анализа чувствительности решения к изменению параметров.

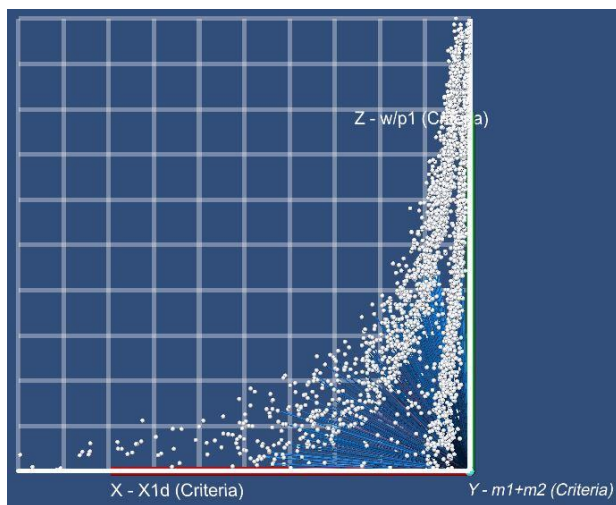
Вынесем суждение типа 2 о близости значений целевых функций к минимальным, то есть к идеальному решению в начале координат. Проводя визу-

альный анализ, отметим наименьшие, то есть более предпочтительные значения целевых функций.

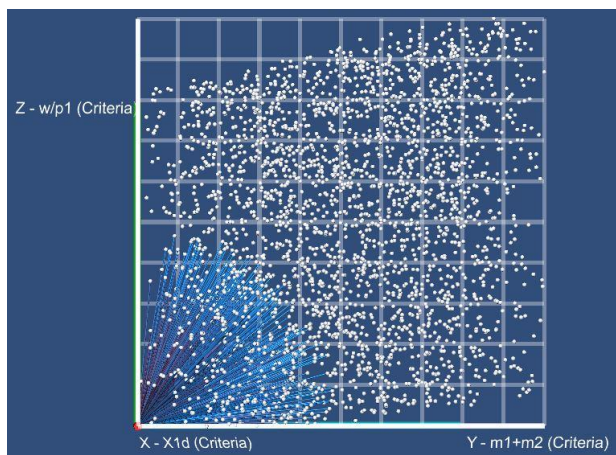
Из рисунка 11 можно увидеть, что:

1) Значения функции X_{10} близки к минимальному – сферы образуют скопление возле плоскости OYZ.

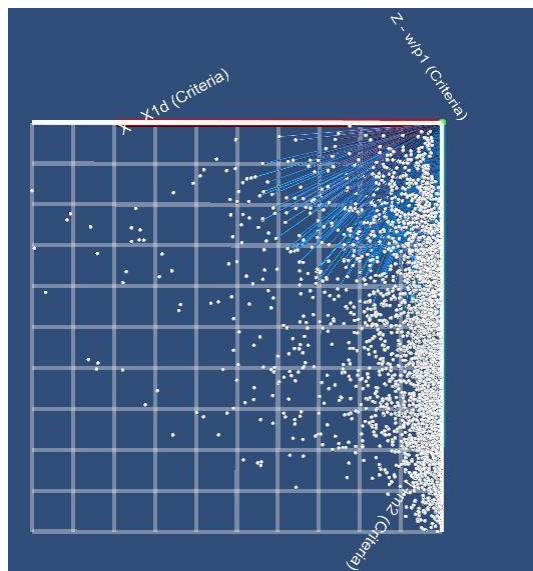
2) Значения функций $\frac{\omega}{p_1}, C_2$ разбросаны по всей области определения, значения целевой функции сильно отличаются от минимального – сферы распределены по всей сцене в профильной проекции.



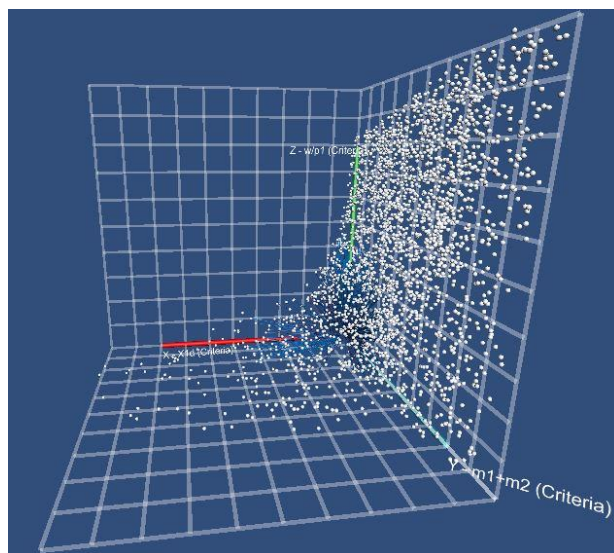
а



б



в



г

Рисунок 11. Сцена со сферами и цилиндрами:

а – фронтальная проекция; б – профильная проекция; в – горизонтальная проекция; г – проекционное изображение

Вынесем суждение типа 1 о схожести и различии положения сфер. Значения двух целевых функций $\frac{\omega}{p_1}, X_{10}$ образуют облако точек. Граница этого облака по форме сильно напоминает гиперболу, из чего можно сделать предложение, что это связано с наложенными на параметрами функциональными ограничениями.

Для функций $\frac{\omega}{p_1}, C_2$ зависимость не наблюдается – сферы распределены практически равномерно.

У функций X_{1d}, C_2 значения распределяются вблизи малых по отношению к нулю значений функции X_{1d} – сферы распределены вдоль одной из осей.

Выделим цветом близкие к нулю сферы в пространстве критериев. Для этого воспользуемся возможностью строить цилиндры до начала координат.

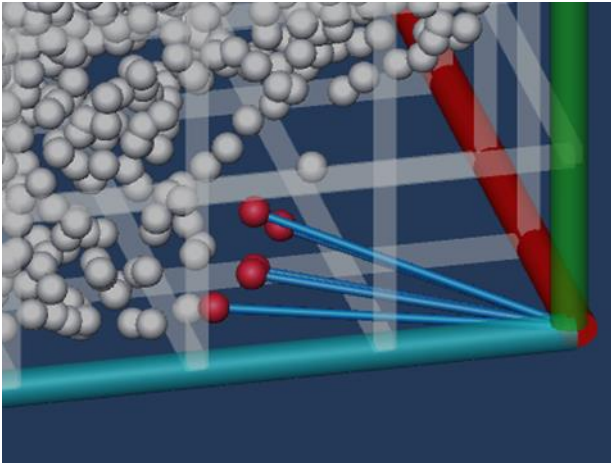


Рисунок 12. Сцена со сферами и цилиндрами.

рейдём в пространство параметров и отметим цветом ближайшие к выделенным сферы.

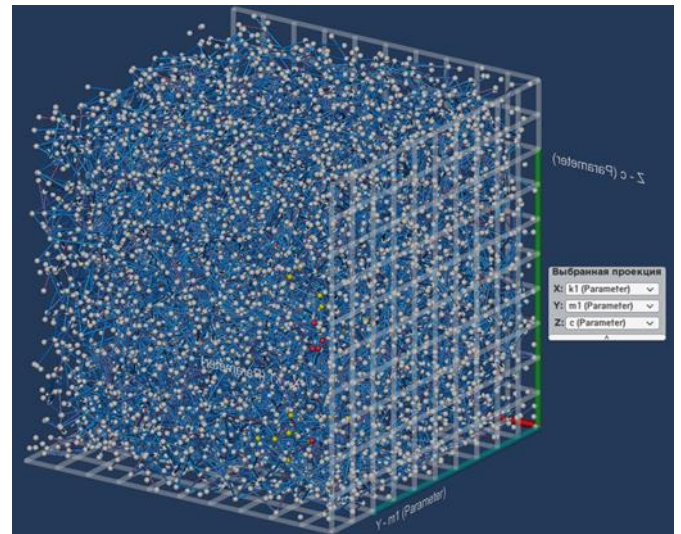


Рисунок 13. Сцена со сферами и цилиндрами.

Перейдём обратно в пространство критериев.

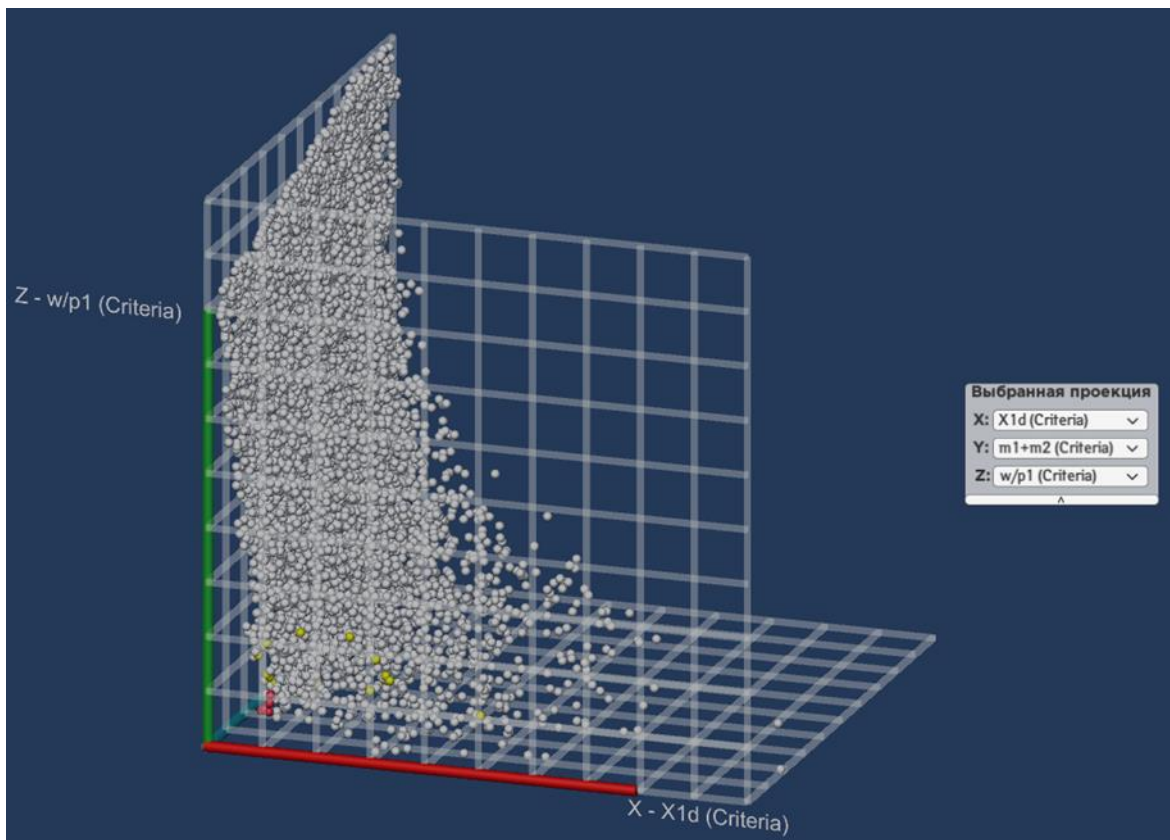


Рисунок 14. Сцена со сферами и цилиндрами.

Вынесем суждение типа 1 о схожести и различии положения сфер для рисунка 14. Сферы, близкие в пространстве параметров к выделенным первоначально сферам, не являются близкими в пространстве критериев. Это означает, что при небольшом изменении параметров, значение целевой функции значительно меняется.

Заключение

В статье были описаны разработка и использование программного средства визуального анализа. Визуальная аналитика применялась в отношении задачи многокритериальной оптимизации, общее представление о которой было приведено вначале. Также была построена модель данных для многокритериальных задач, разработан алгоритм, позволяющий осуществлять анализ этих данных методом визуализации. С помощью реализованной программы был проведён анализ данных задачи, связанной с оптимизацией двухмассовой динамической модели.

Была проанализирована чувствительность решений оптимизационной задачи, показано, что решения являются неустойчивыми: при небольших изменениях параметров модели происходит существенное изменение её оптимизационных критериев. Визуально была выделена совокупность решений задачи, являющаяся оптимальной по агре-

гированному критерию $\sqrt{\sum_{r=1}^q (c_r)^2}$. Также, по форме проекций многомерных облаков точек были сделаны суждения о наличии зависимости между оптимизируемыми критериями.

В дальнейшем планируется развитие в нескольких направлениях. Первое – это расширение возможностей интерактивного взаимодействия пользователя с разработанной программой. А именно, будет добавлена возможность одновременно работать сразу с несколькими трёхмерными проекциями точек в многомерном пространстве, пользователь сможет изменять оптические параметры сцены: раскрашивать сферы в разные

цвета, таким образом наглядно выделяя заинтересовавшие его совокупности. Также будет добавлена функция построения 2D-графики для анализа, например, столбчатых диаграмм частот для рассчитанных значений параметров и критериев.

Второе – включение в программу численных методов, применяемых для решения задач многокритериальной оптимизации, позволяющих без использования визуализации получить потенциально оптимальные решения.

Источники информации

1. Анализ многомерных данных в задачах многопараметрической оптимизации с применением методов визуализации / Бондарев А.Е., Галактионов В.А. // Научная визуализация. – 2012. – том 4 – №2 – С.1-13.

2. Исследование процедур метода анализа иерархий и использование визуализации для проведения анализа чувствительности / В. В. Котова // Научная визуализация. – 2016. – том 8. – №2. – С. 59-84.

3. Использование визуализации при решении дискретных многокритериальных задач методами семейства PROMETHEE / А.Ю. Яковличев, И.Е. Мильман, В.В. Пилюгин // Научная визуализация. – 2016. – том 8. – №3. – С. 78-84.

4. Научная визуализация как метод анализа научных данных / В. В. Пилюгин [и др.] // Научная визуализация. – 2012. – том 4. – №4. – С. 56-70.

5. Использование визуализации при решении задач многокритериального выбора / А.П. Нелюбин [и др.] // Научная визуализация. – 2017. – том 9. – №5. – С. 59-70.

6. Statnikov R. The Parameter Space Investigation Method Toolkit. [Текст] / R. Statnikov, A. Statnikov – Artech House. – 2011. – 214 с,

7. Implementing Sobol's Quasirandom Sequence Generator / Stephen Joe, Frances Y. Kuo // ACM Transactions on Mathematical Software, Vol. 29, No. 1, March 2003, Pages 49–57.

8. Анализ данных о деятельности кредитных организаций с использованием программы интерактивного визуального анализа многомерных данных / И.Е.Мильман [и др.] // Научная визуализация. – 2015. – том 1. – №1 – С. 45 – 64.