Правительство Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Санкт-Петербургский государственный университет»

Кафедра системного программирования

Якшин Алексей Сергеевич

Алгоритмы и программные средства управления проектами с помощью методов тропической математики

Магистерская работа

Допущена к защите.
Зав. кафедрой:
д.ф.-м.н., проф. А.Н. Терехов

Научный руководитель:
д.ф.-м.н., проф. Н.К. Кривулин

Рецензент:
инженер К.К. Смирнов

Санкт-Петербург

2015

SAINT-PETERSBURG STATE UNIVERSITY

Mathematics and Mechanics Faculty

Software Engineering Chair

Alexey Yakshin

Algorithms and software tools for project scheduling by using methods of tropical mathematics

Master Thesis

Admitted for defence.
Head of the chair:
professor A.N. Terekhov

Scientific supervisor:
professor N.K. Krivulin

Reviewer:
engineer K. Smirnov

Saint-Petersburg

2015

Оглавление

Введение

Управление проектами — одна из важнейших инженерных областей в современном мире. В связи со все нарастающей сложностью технических систем стандартные подходы в управлении наукоемкими проектами зачастую неэффективны. Более того, создание графиков и расписаний мало поддается формализации, в связи с чем возникает большое количество рисков, которые могут не позволить проекту завершиться в срок. Цель данной работы состоит в разработке решения для управления проектами с помощью аппарата идемпотентной математики. При декомпозиции основная цель была разделена на следующие задачи:

- Изучение предметной области управления проектами;
- Создание обзора программ управления проектами;
- Изучение идемпотентной(тропической) математики;
- Изучение задач управления проектами, решенных методами тропической математики;
- Разработка программного обеспечения.

Структура данного исследования соответствует поставленным задачам. Первая глава посвящена обзору наиболее распространенных алгоритмов и рассмотрены различные виды прикладных программ для управления проектами. Во второй главе рассматривается идемпотентная алгебра, а также представление задач оптимизации проектов в терминах идемпотентной математики и численные примеры расчета типового проекта. Следующая глава содержит основные моменты реализации рассмотренных алгоритмов в виде программного обеспечения. В заключении представлены выводы, сделанные при выполнении данной работы.

1 Обзор предметной области

1.1 Проект

Проект — временное предприятие, направленное на создание уникального продукта, услуги или результата. Временный характер проектов указывает на определенное начало и окончание. Окончание наступает тогда, когда цели проекта достигнуты или когда проект прекращается в связи с тем, что его цели не будут или не могут быть достигнуты, либо когда в проекте больше нет необходимости. Управление проектом — это приложение знаний, навыков, инструментов и методов к работам проекта для удовлетворения требований, предъявляемых к проекту. Управление проектом осуществляется посредством надлежащего применения и интеграции логически сгруппированных процессов управления. Управление проектом, как правило, включает в себя:

- определение требований;
- реагирование на различные потребности, сомнения и ожидания заинтересованных сторон по мере планирования и выполнения проекта;
- установление и поддержание активных коммуникаций с заинтересованными сторонами;
- уравновешивание конкурирующих ограничений проекта, которые включают в себя, среди прочего:

  содержание,

  качество,

  расписание,

  бюджет,

ресурсы,

риски.

Условия конкретного проекта влияют на ограничения, на которых необходимо сосредоточить внимание руководителю проекта, и которые требуют эффективного применения и управления подходящими процессами управления проектом.

1.2 Сетевой график

Расписание проекта — выход модели расписания, представляющий взаимосвязанные операции с запланированными датами, длительностями, контрольными событиями и ресурсами. Расписание проекта содержит, по меньшей мере, плановый старт и плановый финиш для каждой операции. Если планирование ресурсов проводится на ранней стадии, расписание проекта остается предварительным до подтверждения выделения ресурсов и утверждения расчетных старта и финиша. Обычно этот процесс происходит не позднее, чем будет разработан план управления проектом. Может быть также разработано целевое расписание проекта с определенным целевым стартом и финишем для каждой операции. Расписание проекта может быть представлено в укрупненном виде, иногда называемом укрупненным расписанием или расписанием контрольных событий, или же в детальном виде. [2]

1.3 Методы управления длительностью проекта

Существует несколько подходов к управлению временем проекта, основными являются:

- График Ганта — самая распространенная диаграмма, которая состоит из полос, ориентированных вдоль оси времени. Каждая полоса на диаграмме представляет отдельную задачу в составе проекта (вид работы), её концы — моменты начала и завершения работы, её протяженность — длительность работы. Вертикальной осью диаграммы служит перечень задач. Кроме того, на диаграмме могут быть отмечены совокупные задачи, проценты завершения, указатели последовательности и зависимости работ, метки ключевых моментов (вехи), метка текущего момента времени и др. Ключевым понятием диаграммы Ганта является «Веха» — метка значимого момента в ходе выполнения работ, общая граница двух или более задач. Вехи позволяют наглядно отобразить необходимость синхронизации, последовательности в выполнении различных работ. Вехи, как и другие границы на диаграмме, не являются календарными датами. Сдвиг вехи приводит к сдвигу всего проекта, поэтому диаграмма Ганта не является, строго говоря, графиком работ, что является одним из основных её недостатков. Кроме того, диаграмма Ганта не отображает значимости или ресурсоемкости работ и их сущности (области действия). Для крупных проектов диаграмма Ганта становится чрезмерно тяжеловесной и теряет всякую наглядность.

- Метод критического пути (англ. CPM — Critical Path Method) позволяет составить план реализации проекта по следующим параметрам: список всех работ для завершения проекта, время, которое требуется для их завершения, связь между работами и логически завершаемые точки — вехи. Вводится такое понятие как критический путь — последовательность операций, наиболее важных для соблюдения установленного срока реализации проекта. Критический путь определяется в процессе анализа длительности выполнения операций в сети. Такой анализ основан на взаимосвязях операций и информации о длительности их выполнения. Критический путь образуется последовательностью операций, объединяющих события с наименьшими резервами времени. Критический путь — самая длинная последовательность операций в сети, её длительность — это срок завершения проекта в целом. Данный метод используются в большинстве программ для управления проектами: Microsoft Project, Oracle Primavera и многими другими.

- Метод оценки и анализа проектов (англ. PERT — Project Evaluation and Review Technique) — способ анализа задач, необходимых для выполнения проекта. В особенности, анализа времени, которое требуется для выполнения каждой отдельной задачи, а также определение минимального необходимого времени для выполнения всего проекта. Данный метод оперирует следующими параметрами:

  PERT Событие (PERT event): точка, отмечающая начало или окончание одной или более активностей;

  Предшествующее Событие (predecessor event): событие, которое непосредственно предшествует некоторому другому событию, без иных промежуточных событий;

  Последующее событие (successor event): событие, которое непосредственно следует за некоторым иным событием, без иных промежуточных событий;

  PERT активность (PERT activity): реальное выполнение задачи, которое потребляет время и требует ресурсов;

  Пессимистическое время (pessimistic time) (P): максимально возможное время, требующееся для выполнения задачи, в предположении, что все происходит наихудшим образом (исключая крупные катастрофы);

Критический путь (critical path): Длиннейший маршрут на пути от начального до финального события;

Критическая активность (critical activity): Активность, общее проскальзывание которой равно нулю;

Быстрый проход (fast tracking): метод сжатия расписания проекта, изменяющий логику сети путем наложения друг на друга фаз, которые в обычной ситуации выполнялись бы последовательно (например, наложение фаз проектирования и строительства), или для параллельного выполнения запланированных операций;

Наиболее вероятное время (most likely time) (M): оценка времени, требующегося для выполнения задачи, в предположении, что все происходит как обычно;

Ожидаемое время (expected time) (TE): лучшая оценка времени, требуемого для выполнения задачи, учитывая, что вещи не всегда происходят как обычно. (Ожидаемое среднее время выполнения задачи, если она будет повторяться многократно);

Проскальзывание или провисание: мера дополнительного времени и ресурсов доступных для выполнения работы;

Оптимистическое время (optimistic time) (O): минимальное возможное время для выполнения задачи, в предположении, что все происходит наилучшим образом;

PERT была разработана, главным образом, для упрощения планирования на бумаге и составления графиков больших и сложных проектов. PERT предназначена для очень масштабных, единовременных, сложных, нерутинных проектов. Техника подразумевала наличие неопределённости, давая возможность разработать рабочий график проекта без точного знания деталей и необходимого времени для всех его составляющих.

- Метод графической оценки и анализа (англ. GERT — Graphical Evaluation and Review Technique) — альтернативный вероятностный метод сетевого планирования, применяющийся в случаях организации работ, когда последующие задачи могут начинаться после завершения только некоторого числа из предшествующих задач, однако, не все задачи, представленные на сетевой модели, должны быть выполнены для завершения проекта. Основу применения метода GERT составляет использование альтернативных сетей, называемых GERT-сетями. Они позволяют более адекватно задавать сложные процессы строительного производства в тех случаях, когда затруднительно или невозможно (по объективным причинам) однозначно определить, какие именно работы и в какой последовательности должны быть выполнены для достижения цели проекта (то есть существует многовариантность реализации проекта). Расчёт GERT-сетей, моделирующих реальные процессы, чрезвычайно сложен, однако программное обеспечение для вычисления сетевых моделей такого типа в настоящее время не распространено.

1.4 Программное обеспечение для управления проектами

Программное обеспечение для управления проектами — комплексное программное обеспечение, включающее в себя приложения для планирования задач, составления расписания, контроля цены и управления бюджетом, распределения ресурсов, совместной работы, общения, быстрого управления, документирования и администрирования системы, которое, используются совместно для управления крупными проектами. Существует два основных класса подобных систем:

- Настольные — программное обеспечение установлено на персональном компьютере каждого пользователя. Такие приложения обычно позволяют сохранять информацию в файл, который в дальнейшем может быть выложен в общий доступ для других пользователей или же данные хранятся в центральной базе данных.
- Веб-приложения — программное обеспечение является веб-приложением, доступ к которому осуществляется с помощью браузера.

Рассмотрим несколько представителей для каждого из классов:

- Microsoft Project — стандарт де-факто в области управления проектами. Позволяет создавать планы, распределять ресурсы по задачам, отслеживать прогресс и анализировать объём работ. Программа способна создавать расписания критического пути. Расписания могут быть составлены с учётом используемых ресурсов. Цепочка визуализируется при помощи диаграммы Ганта. Приложение является настольным, поэтому его использование возможно только на персональных компьютерах.
- Oracle Primavera — большой набор связанных приложений, который позволяет: выбрать нужное сочетание стратегических проектов, обеспечить корпоративное управление проектом, улучшить процессы и методы, улучшить совместную работу над проектом, измерить прогресс в достижении целей, связать проекты с выбранной предприятием стратегией. Данное программное обеспечение используется для управления

большими проектами, по масштабам сравнимыми со строительством ядерного реактора, поэтому целевая аудитория Primavera — корпоративный сегмент.

- Basecamp от 37signals — онлайн-инструмент для управления проектами, совместной работы и постановки задач, подходит для небольших проектов, имеет скромное, но полезное функциональное наполнение: назначение и отслеживание задач, загрузка, категоризация и отслеживание версий файлов, ведение расписания и управление ключевыми точками проекта, отслеживание потраченного времени. Возможна расширяемость сторонними дополнениями, целевая аудитория — проектные команды состоящие из 3-5 человек.

- Redmine — серверное веб-приложение для управления проектами и задачами (в том числе для отслеживания ошибок), предоставляет следующие возможности: ведение нескольких проектов, диаграммы Ганта и календарь, учёт временных затрат, настраиваемые произвольные поля для инцидентов, временных затрат, проектов и пользователей и некоторые другие. Является универсальным инструментом с открытым исходным кодом, может быть дополнен сторонними расширениями, подходит для средних проектов. Является свободно распространяемым программным обеспечением, создателем является Jean-Philippe Lang.

2 Задачи сетевого планирования управления проектами

2.1 Задача минимального общего времени выполнения проекта

Для детального рассмотрения, была выбрана задача минимального общего времени выполнения проекта, так как она представляет наибольший интерес в области управления проектами(для дополнительной информации о задаче стоит обратиться к [3, 4, 14, 19–21]). Рассмотрим проект, который состоит из n работ, находящихся под временными ограничениями типа старт-финиш, ранний старт и поздний старт. Ограничения типа старт-финиш обозначают нижний лимит для разрешенного временного лага между началом одной работы и завершением другой. Работы должны быть завершены настолько быстро, насколько это возможно при ограничениях типа старт-финиш. Ограничения типа ранний старт и поздний старт означают, соответственно, самое раннее и позднее возможное время начала и завершения для каждой работы. Ниже рассмотрим задачу, которая имеет только ограничения типа старт-финиш, и затем расширим результат общей задачи с наложением ограничений типа старт-финиш, ранний старт и поздний старт.

Для каждой работы $i = 1, \ldots, n$, введем время начала x_i и время завершения — y_i. Пусть c_{ij} — минимальное время задержки между началом деятельности $j = 1, \ldots, n$ и завершением работы i. Если c_{ij} не задано для некоторого j будем считать равным $c_{ij} = -\infty$. Время завершения работы i должно удовлетворять отношениям ограничений типа старт-финиш [3]

$$x_j + c_{ij} \leq y_i, \quad j = 1, \ldots, n,$$

где как минимум одно неравенство является равенством. Совмещение отношений дает

$$y_i = \max_{1 \le j \le n}(x_j + c_{ij}), \quad i = 1, \ldots, n. \tag{1}$$

Общее время завершения есть время между самым ранним началом и самым поздним завершением работ в проекте. Может быть записано в форме

$$\max_{1 \le i \le n} y_i - \min_{1 \le i \le n} x_i = \max_{1 \le i \le n} y_i + \max_{1 \le i \le n}(-x_i).$$

После подстановки y_i определенной в (1), задача создания графика с минимальным общим временем выполнения проекта может быть сформулирована в стандартной форме как: дано c_{ij} для $i, j = 1, \ldots, n$, найти $x_1, \ldots, x_n$, решающие задачу:

$$\text{минимизировать} \quad \max_{1 \le i \le n} \max_{1 \le j \le n}(x_j + c_{ij}) + \max_{1 \le i \le n}(-x_i).$$

2.2 Обзор тропической математики

Рассмотрим основные определения и результаты тропической математики. Обзор основан на результатах, описанных в [3,14,15,22]. Для дополнительной информации можно обратиться к [5–9].

Пусть $\mathbb{X}$ есть множество с двумя ассоциативными и коммутативными операциями, $\oplus$ (сложение) и $\otimes$ (умножение), аддитивным и мультипликативным нейтральными элементами, $\mathbb{0}$ (ноль) и $\mathbb{1}$ (единица). Сложение идемпотентно: $x \oplus x = x$ для каждого $x \in \mathbb{X}$. Умножение дистрибутивно относительно сложения и обратимо, т.е. для каждого элемента существует обратный $x^{-1} \otimes x = \mathbb{1}$. Систему $(\mathbb{X}, \oplus, \otimes, \mathbb{0}, \mathbb{1})$ называют идемпотентным полуполем. Полуполе линейно упорядочено с помощью порядка, который согласован с частичным порядком, порожденным идемпотентным сложением: $x \le y$ тогда и только тогда, когда $x \oplus y = y$. Более того, полуполе считается алгебраически полным, т.е. для любого $a \neq \mathbb{0}$ и целого m существует решение уравнения $x^m = a$.

Примеры полуполей: $\mathbb{R}_{\max,+} = (\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0)$, $\mathbb{R}_{\min,+} = (\mathbb{R} \cup \{+\infty\}, \min, +, +\infty, 0)$, $\mathbb{R}_{\max,\times} = (\mathbb{R}_+ \cup \{0\}, \max, \times, 0, 1)$ и $\mathbb{R}_{\min,\times} = (\mathbb{R}_+ \cup \{+\infty\}, \min, \times, +\infty, 1)$, где $\mathbb{R}$ - множество вещественных чисел и $\mathbb{R}_+ = \{x \in \mathbb{R} | x > 0\}$.

Знак умножения $\otimes$ для простоты будем опускать.

Пусть $\mathbb{X}^{m\times n}$ есть множество матриц с m строками и n столбцами над $\mathbb{X}$. Сложение и умножение и скалярное умножение определены стандартными правилами со скалярными операциями $\oplus$ и $\otimes$ взятыми вместо обычных сложения и умножения . Матрица со всеми элементами, равными $\mathbb{0}$, называется нулевой матрицей.

Рассмотрим множество $\mathbb{X}^{n\times n}$ квадратных матриц порядка n. Матрица с $\mathbb{1}$ на главной диагонали и $\mathbb{0}$ на других позициях называется единичной $\boldsymbol{I}$.

Для любой матрицы $\boldsymbol{A} = (a_{ij}) \in \mathbb{X}^{n\times n}$ идемпотентные аналоги следа, нормы и спектрального радиуса даны соответственно

$$\operatorname{tr}\boldsymbol{A} = \bigoplus_{i=1}^{n} a_{ii}, \qquad \|\boldsymbol{A}\| = \bigoplus_{i=1}^{n}\bigoplus_{j=1}^{n} a_{ij}, \qquad \lambda = \bigoplus_{m=1}^{n} \operatorname{tr}^{1/m}(\boldsymbol{A}^m).$$

Множество вектор-столбцов размера n над $\mathbb{X}$ обозначаются $\mathbb{X}^n$. Вектор со всеми элементами равными $\mathbb{0}$ называется нулевым и обозначается как $\mathbf{0}$. Вектор называется регулярным, если не имеет нулевых элементов. Для любого ненулевого вектора $\boldsymbol{x} = (x_i) \in \mathbb{X}^n$, мультипликативное сопряженное транспонирование дает вектор-строку $\boldsymbol{x}^- = (x_i^-)$ с элементами $x_i^- = x_i^{-1}$, если $x_i \neq \mathbb{0}$, и $x_i^- = \mathbb{0}$ в ином случае. Сопряженное транспонирование обладает полезными свойствами, которые легко проверить. Для любого ненулевого вектора $\boldsymbol{x}$ мы имеем уравнение $\boldsymbol{x}^-\boldsymbol{x} = \mathbb{1}$. Для регулярных векторов $\boldsymbol{x}$ и $\boldsymbol{y}$ одинаковой размерности выполняется неравенство $\boldsymbol{x}\boldsymbol{y}^{-} \geq (\boldsymbol{x}^{-}\boldsymbol{y})^{-1}\boldsymbol{I}$.

Для любой матрицы $\boldsymbol{A}$ выполняется $\|\boldsymbol{A}\| = \mathbf{1}^T\boldsymbol{A}\mathbf{1}$, где $\mathbf{1} = (\mathbb{1}, \ldots, \mathbb{1})^T$.

Для данного вектора $\boldsymbol{a} \in \mathbb{X}^n$ и числа $d \in \mathbb{X}$ рассмотрим задачу нахождения регулярных векторов $\boldsymbol{x} \in \mathbb{X}^n$ которые удовлетворяют решения уравнения

$$\boldsymbol{a}^T\boldsymbol{x} = d. \tag{2}$$

Лемма 1 ([22])**.** *Пусть* $\boldsymbol{a} = (a_i)$ *есть регулярный вектор и* $d \neq \mathbb{0}$ *— число. Тогда, решение равенства* (2) *образует семейство решений, каждое из которых определено для* $k = 1, \ldots, n$ *как набор векторов* $\boldsymbol{x} = (x_i)$ *с компонентами*

$$x_k = a_k^{-1}d, \qquad x_i \leq a_i^{-1}d, \quad i \neq k.$$

Рассмотрим задачу: дана матрица $\boldsymbol{A} \in \mathbb{X}^{n\times n}$, найти регулярные векторы $\boldsymbol{x} \in \mathbb{X}^n$, которые минимизируют выражение

$$\boldsymbol{x}^{-}\boldsymbol{A}\boldsymbol{x}. \tag{3}$$

Для решения задачи используем оператор под названием астерат(известный как звезда Клини), который преобразует любую матрицу $\boldsymbol{A} \in \mathbb{X}^{n\times n}$ в матрицу

$$\boldsymbol{A}^{*} = \boldsymbol{I} \oplus \boldsymbol{A} \oplus \cdots \oplus \boldsymbol{A}^{n-1}.$$

Лемма 2 ([15, 17, 18])**.** *Пусть* $\boldsymbol{A}$ *— матрица со спектральным радиусом* $\lambda > \mathbb{0}$. *Тогда, ответ в задаче* (3) - λ *и все регулярные решения могут быть получены как*

$$\boldsymbol{x} = (\lambda^{-1}\boldsymbol{A})^{*}\boldsymbol{u}, \qquad \boldsymbol{u} > \mathbf{0}.$$

Найдем решение для задачи с ограничениями (3). Дана матрица $\boldsymbol{A} \in \mathbb{X}^{n\times n}$ и векторы $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{X}^n$, найти регулярные векторы $\boldsymbol{x} \in \mathbb{X}^n$, которые минимизируют выражение

$$
\begin{aligned}
& \boldsymbol{x}^{-}\boldsymbol{A}\boldsymbol{x}, \\
\text{при условии} \quad & \boldsymbol{g} \leq \boldsymbol{x} \leq \boldsymbol{h}.
\end{aligned} \tag{4}
$$

Теорема 3 ([18])**.** *Пусть $\boldsymbol{A}$ — матрица со спектральным радиусом $\lambda > \mathbb{0}$ и $\boldsymbol{h}$ — регулярный вектор, $\boldsymbol{h}^{-}\boldsymbol{g} \leq \mathbb{1}$. Тогда — минимальное значение задачи* (4)

$$\theta = \lambda \oplus \bigoplus_{m=1}^{n} (\boldsymbol{h}^{-}\boldsymbol{A}^{m}\boldsymbol{g})^{1/m},$$

все регулярные решения могут быть найдены как

$$\boldsymbol{x} = (\theta^{-1}\boldsymbol{A})^{*}\boldsymbol{u}, \qquad \boldsymbol{g} \leq \boldsymbol{u} \leq (\boldsymbol{h}^{-}(\theta^{-1}\boldsymbol{A})^{*})^{-}.$$

Рассмотрим многомерную задачу оптимизации без ограничений, сформулированную в терминах тропической математики и затем расширим решение для задачи с ограничениями.

Даны векторы $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{X}^{n}$, первая задача заключается в нахождении регулярных векторов $\boldsymbol{x} \in \mathbb{X}^{n}$, которые

$$\text{минимизировать} \quad \boldsymbol{q}^{-}\boldsymbol{x}\boldsymbol{x}^{-}\boldsymbol{p}. \tag{5}$$

Рассмотрим решение, основанное на происхождении нижней границы целевой функции, и решения уравнения для нахождения всех векторов, которые образуют границу.

Теорема 4 ([3])**.** *Пусть $\boldsymbol{p}$ есть ненулевой вектор и $\boldsymbol{q}$ — регулярный вектор. Тогда, минимальное значение задачи* (5) *есть*

$$\Delta = \boldsymbol{q}^{-}\boldsymbol{p}, \tag{6}$$

и все регулярные решения задачи даны как

$$\alpha\boldsymbol{p} \leq \boldsymbol{x} \leq \alpha\Delta\boldsymbol{q}, \qquad \alpha > \mathbb{0}. \tag{7}$$

Для того, чтобы предложить другое решение задачи (5), для начала, преобразуем целевую функцию в эквивалентную форму

$$\boldsymbol{q}^{-}\boldsymbol{x}\boldsymbol{x}^{-}\boldsymbol{p} = \boldsymbol{x}^{-}\boldsymbol{p}\boldsymbol{q}^{-}\boldsymbol{x},$$

и затем рассмотрим задачу

$$\text{минимизировать} \quad \boldsymbol{x}^{-}\boldsymbol{p}\boldsymbol{q}^{-}\boldsymbol{x}. \tag{8}$$

Задача станет частным случаем задачи (3) с $\boldsymbol{A} = \boldsymbol{p}\boldsymbol{q}^{-}$, и поэтому может быть решена с помощью Леммы 2.

Теорема 5 ([3]). *Пусть* $\boldsymbol{p}$ *— ненулевой вектор и* $\boldsymbol{q}$ *— регулярный вектор. Тогда, минимальное значение задачи* (5) *есть*

$$\Delta = \boldsymbol{q}^{-}\boldsymbol{p},$$

и все регулярные решения задачи могут быть получены как

$$\boldsymbol{x} = (\boldsymbol{I} \oplus \Delta^{-1}\boldsymbol{p}\boldsymbol{q}^{-})\boldsymbol{u}, \qquad \boldsymbol{u} > \boldsymbol{0}. \tag{9}$$

2.3 Задача оптимизации с ограничениями

Добавим нижние и верхние ограничения на допустимые решения задачи. Даны векторы $\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{g}, \boldsymbol{h} \in \mathbb{X}^n$, рассмотрим задачу, в которой нужно найти все регулярные векторы $\boldsymbol{x} \in \mathbb{X}^n$, которые удовлетворяют условиям:

$$\begin{aligned} &\text{минимизировать} && \boldsymbol{q}^{-}\boldsymbol{x}\boldsymbol{x}^{-}\boldsymbol{p}, \\ &\text{при условии} && \boldsymbol{g} \leq \boldsymbol{x} \leq \boldsymbol{h}. \end{aligned} \tag{10}$$

Теорема 6 ([3]). *Пусть* $\boldsymbol{p}$ *есть ненулевой вектор,* $\boldsymbol{q}$ *— регулярный вектор, и* $\boldsymbol{h}$ *— регулярный вектор, такой как* $\boldsymbol{h}^{-}\boldsymbol{g} \leq \mathbb{1}$. *Тогда, минимальное значение задачи* (10) *эквивалентно*

$$\theta = \boldsymbol{q}^{-}(\boldsymbol{I} \oplus \boldsymbol{g}\boldsymbol{h}^{-})\boldsymbol{p}, \tag{11}$$

и все регулярные решения задачи могут быть получены как

$$\boldsymbol{x} = (\boldsymbol{I} \oplus \theta^{-1}\boldsymbol{p}\boldsymbol{q}^{-})\boldsymbol{u}, \qquad \boldsymbol{g} \leq \boldsymbol{u} \leq (\boldsymbol{h}^{-}(\boldsymbol{I} \oplus \theta^{-1}\boldsymbol{p}\boldsymbol{q}^{-}))^{-}. \tag{12}$$

2.4 Решение задачи общего минимального выполнения проекта с помощью тропической математики

Задача 1 в терминах полуполя $\mathbb{R}_{\max,+}$, может быть представлена в форме

$$\text{минимизировать} \quad \bigoplus_{i=1}^{n}\bigoplus_{j=1}^{n} c_{ij}x_j \bigoplus_{k=1}^{n} x_k^{-1}.$$

Введем матрицу $\boldsymbol{C} = (c_{ij})$ и вектор $\boldsymbol{x} = (x_i)$. Используя данное обозначение, запишем задачу в векторной форме

$$\text{минимизировать} \quad \mathbf{1}^T\boldsymbol{C}\boldsymbol{x}\boldsymbol{x}^{-}\mathbf{1}. \tag{13}$$

Последняя задача есть особый случай задачи (5), где $\boldsymbol{p} = (\mathbb{1},\ldots,\mathbb{1})^T = \mathbf{1}$ и $\boldsymbol{q}^{-} = \mathbf{1}^T\boldsymbol{C}$.

Как следствие Теорем 4 и 5, получим следующий результат.

Теорема 7 ([3])**.** *Пусть* $\boldsymbol{C}$ *— регулярная по строкам матрица. Тогда, минимальное значение в задаче* (13) *эквивалентно*

$$\Delta = \mathbf{1}^T\boldsymbol{C}\mathbf{1} = \|\boldsymbol{C}\|,$$

регулярные решения задачи задаются следующим образом:

$$\alpha\mathbf{1} \leq \boldsymbol{x} \leq \alpha\Delta(\mathbf{1}^T\boldsymbol{C})^{-}, \qquad \alpha \in \mathbb{R}, \tag{14}$$

или эквивалентно

$$\boldsymbol{x} = (\boldsymbol{I} \oplus \Delta^{-1}\mathbf{1}\mathbf{1}^T\boldsymbol{C})\boldsymbol{u}, \qquad \boldsymbol{u} \in \mathbb{R}^n. \tag{15}$$

Вернемся к общей задаче и добавим ограничения типа ранний старт и поздний старт. Пусть g_i есть самое раннее время и h_i есть самое позднее время для начала работы i. Ограничения на ранний старт и поздний финиш заключают в себе неравенство

$$g_i \leq x_i \leq h_i, \quad i = 1, \ldots, n.$$

В векторной форме $\boldsymbol{g} = (g_i)$ и $\boldsymbol{h} = (h_i)$ расширим задачу без ограничений (13) до задачи с ограничениями

$$\begin{aligned} &\text{минимизировать} && \mathbf{1}^T \boldsymbol{C}\boldsymbol{x}\boldsymbol{x}^- \mathbf{1}, \\ &\text{при условии} && \boldsymbol{g} \leq \boldsymbol{x} \leq \boldsymbol{h}. \end{aligned} \tag{16}$$

При применении Теоремы (6) с $\boldsymbol{p} = \mathbf{1}$ и $\boldsymbol{q}^- = \mathbf{1}^T \boldsymbol{C}$, приходим к следующему результату.

Теорема 8 ([3])**.** *Пусть* $\boldsymbol{C}$ *— строка-регулярная матрица и* $\boldsymbol{h}$ *— вектор такой как* $\boldsymbol{h}^- \boldsymbol{g} \leq \mathbb{1}$. *Тогда, минимальное значение в задаче* (16) *эквивалентно*

$$\theta = \mathbf{1}^T \boldsymbol{C}(\boldsymbol{I} \oplus \boldsymbol{g}\boldsymbol{h}^-)\mathbf{1} = \|\boldsymbol{C}(\boldsymbol{I} \oplus \boldsymbol{g}\boldsymbol{h}^-)\|,$$

регулярные решения задачи задаются следующим образом:

$$\boldsymbol{x} = (\boldsymbol{I} \oplus \theta^{-1} \mathbf{1}\mathbf{1}^T \boldsymbol{C})\boldsymbol{u}, \qquad \boldsymbol{g} \leq \boldsymbol{u} \leq (\boldsymbol{h}^-(\boldsymbol{I} \oplus \theta^{-1} \mathbf{1}\mathbf{1}^T \boldsymbol{C}))^-.$$

2.5 Численный пример

Проиллюстрируем результаты на примере проекта, который состоит из $n = 3$ работ со следующими ограничениями

$$\boldsymbol{C} = \begin{pmatrix} 4 & 2 & -\infty \\ 1 & 3 & -1 \\ 0 & 2 & 2 \end{pmatrix}, \qquad \boldsymbol{g} = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}, \qquad \boldsymbol{h} = \begin{pmatrix} 4 \\ 4 \\ 2 \end{pmatrix}.$$

Начнем с задачи (13), где не определены ограничения раннего начала и ограничения типа поздний старт. Применение Теоремы 7 даст нам минимальное общее время выполнения проекта.

$$\Delta = \|\boldsymbol{C}\| = 4.$$

Для представления решения задачи в форме(14), вычислим векторы

$$\mathbf{1}^T\boldsymbol{C} = \begin{pmatrix} 4 & 3 & 2 \end{pmatrix}, \qquad (\mathbf{1}^T\boldsymbol{C})^- = \begin{pmatrix} -4 \\ -3 \\ -2 \end{pmatrix}, \qquad \Delta(\mathbf{1}^T\boldsymbol{C})^- = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix},$$

и тогда запишем решение

$$\alpha \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \leq \boldsymbol{x} \leq \alpha \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}, \qquad \alpha \in \mathbb{R}.$$

В скалярной форме, записанной в терминах стандартных операций, решение дается как

$$x_1 = \alpha \qquad \alpha \leq x_2 \leq \alpha + 1, \qquad \alpha \leq x_3 \leq \alpha + 2, \qquad \alpha \in \mathbb{R}.$$

Кроме того, мы получаем решения, предлагаемые в (15). После вычисления матриц

$$\Delta^{-1}\mathbf{1}\mathbf{1}^T\boldsymbol{C} = \begin{pmatrix} 0 & -1 & -2 \\ 0 & -1 & -2 \\ 0 & -1 & -2 \end{pmatrix}, \qquad \boldsymbol{I} \oplus \Delta^{-1}\mathbf{1}\mathbf{1}^T\boldsymbol{C} = \begin{pmatrix} 0 & -1 & -2 \\ 0 & 0 & -2 \\ 0 & -1 & 0 \end{pmatrix},$$

получим решение

$$\boldsymbol{x} = \begin{pmatrix} 0 & -1 & -2 \\ 0 & 0 & -2 \\ 0 & -1 & 0 \end{pmatrix} \boldsymbol{u}, \qquad \boldsymbol{u} \in \mathbb{R}^3.$$

В стандартной нотации предположим, что u_1, u_2, u_3 действительные числа, тогда

$$x_1 = \max(u_1, u_2 - 1, u_3 - 2), \qquad x_2 = \max(u_1, u_2, u_3 - 2),$$
$$x_3 = \max(u_1, u_2 - 1, u_3).$$

Предположим теперь, что есть ограничения типа старт-финиш ограничения на ранний старт и поздний старт остаются так же в силе. Для применения Теоремы 8, для начала, запишем матрицы

$$\boldsymbol{g}\boldsymbol{h}^{-} = \begin{pmatrix} -1 & -1 & 1 \\ -2 & -2 & 0 \\ -3 & -3 & -1 \end{pmatrix}, \qquad \boldsymbol{I} \oplus \boldsymbol{g}\boldsymbol{h}^{-} = \begin{pmatrix} 0 & -1 & 1 \\ -2 & 0 & 0 \\ -3 & -3 & 0 \end{pmatrix},$$

$$\boldsymbol{C}(\boldsymbol{I} \oplus \boldsymbol{g}\boldsymbol{h}^{-}) = \begin{pmatrix} 4 & 3 & 5 \\ 1 & 3 & 3 \\ 0 & 2 & 2 \end{pmatrix},$$

и тогда получим минимальное время выполнение проекта

$$\theta = \|\boldsymbol{C}(\boldsymbol{I} \oplus \boldsymbol{g}\boldsymbol{h}^{-})\| = 5.$$

Кроме того, мы успешно получили матрицы

$$\theta^{-1}\mathbf{1}\mathbf{1}^T\boldsymbol{C} = \begin{pmatrix} -1 & -2 & -3 \\ -1 & -2 & -3 \\ -1 & -2 & -3 \end{pmatrix}, \qquad \boldsymbol{I} \oplus \theta^{-1}\mathbf{1}\mathbf{1}^T\boldsymbol{C} = \begin{pmatrix} 0 & -2 & -3 \\ -1 & 0 & -3 \\ -1 & -2 & 0 \end{pmatrix},$$

и вектор

$$\boldsymbol{h}^{-}(\boldsymbol{I} \oplus \theta^{-1}\mathbf{1}\mathbf{1}^T\boldsymbol{C}) = \begin{pmatrix} -3 & -4 & -2 \end{pmatrix}.$$

Решение, полученное с помощью Теоремы 8 , приобретает форму

$$\boldsymbol{x} = \begin{pmatrix} 0 & -2 & -3 \\ -1 & 0 & -3 \\ -1 & -2 & 0 \end{pmatrix}\boldsymbol{u}, \qquad \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix} \leq \boldsymbol{u} \leq \begin{pmatrix} 3 \\ 4 \\ 2 \end{pmatrix}$$

Скалярное представление в обычной нотации дает следующие равенства:

$$x_1 = \max(u_1, u_2 - 2, u_3 - 3), \qquad x_2 = \max(u_1 - 1, u_2, u_3 - 3),$$
$$x_3 = \max(u_1 - 1, u_2 - 2, u_3),$$

где числа u_1, u_2 and u_3 удовлетворяют условиям

$$u_1 = 3, \qquad 2 \leq u_2 \leq 4, \qquad 1 \leq u_3 \leq 2.$$

Объединив условия из равенств, указанных выше, получим решение

$$x_1 = 3, \qquad 2 \leq x_2 \leq 4, \qquad x_3 = 2.$$

3 Разработка программного обеспечения

3.1 Архитектура разработанной системы

После анализа приложений управления проектами был сделан выбор в пользу создания веб-приложения, доступного всем платформам, способным использовать современные веб-браузеры. Данное решение позволит увеличить потенциальную целевую аудиторию и избавит от проблемы совместимости. Диаграмма деятельности представлена на рисунке1 .

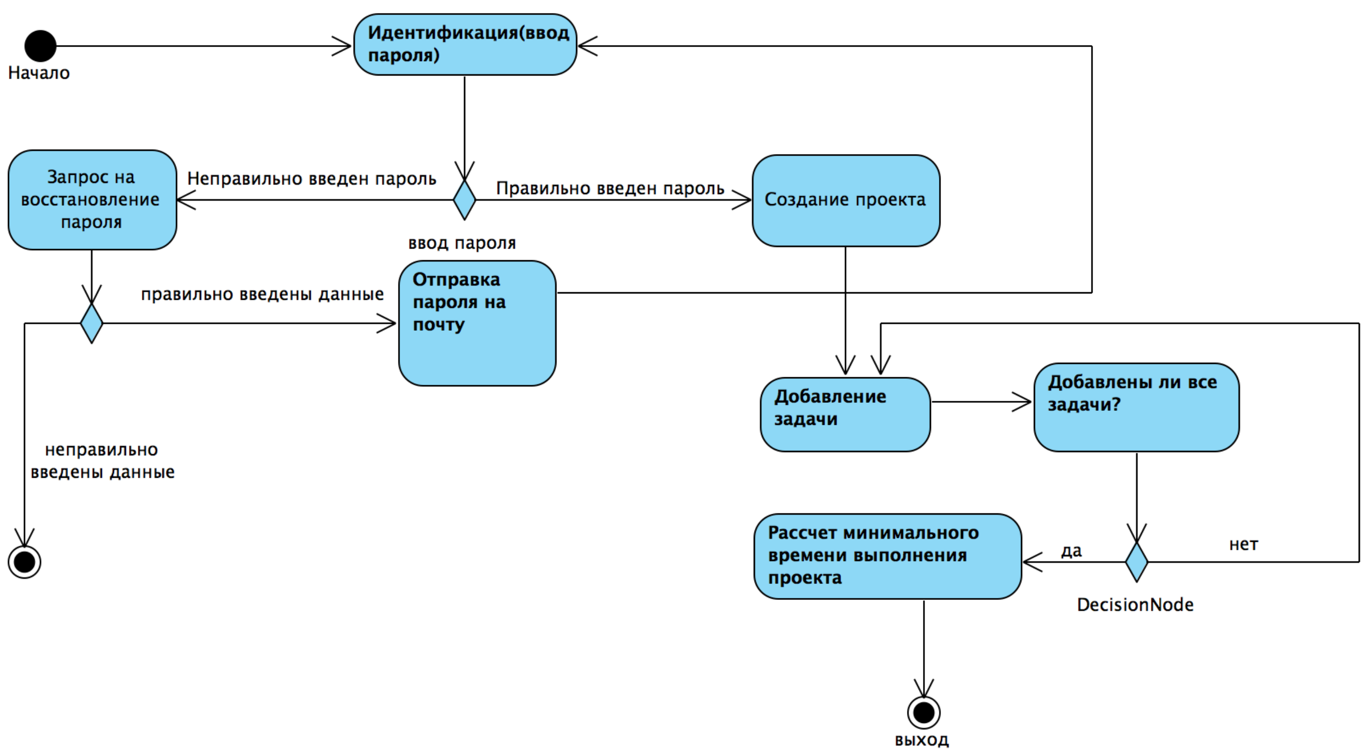

Рис. 1: Диаграмма деятельности

Поскольку выбор сделан в пользу создания веб-приложения, главным языком реализации был выбран JavaScript, который позволит легко перенести вычисления с сервера на клиент и обратно, в зависимости от текущих нужд, и использовать библиотеку в робототехнике. Недостаток использования данного языка в том, что он не имеет в себе механизмов перегрузки операторов, что облегчило бы реализацию библиотеки, однако, его распространенность и гибкость дает нам более весомое преимущество. Возможность переноса вычислений с сервера на клиентское устройство позволяет использовать облачные вычисления с наименьшими денежными затратами и сделать выбор в пользу распространения приложения как сервиса (англ. SaaS — Software as a Service). Расположение решения в облаке дает нам как финансовую масштабируемость, так и техническую, нам требуется платить только за реально использованное процессорное время и дисковое пространство, при увеличении клиентов увеличение процессорных и дисковых мощностей произойдет автоматически, точно так же, как и затраты на сервер повысятся вместе с прибылью от притока новых пользователей. Другое обязательное требование к разрабатываемой системе - адаптивность, универсальность отображения веб-приложения на всем спектре существующих устройств, поддерживающих веб-браузеры, — реализована с помощью использования последней версии языка HTML (англ. HyperText Markup Language, version 5 — язык гипертекстовой разметки) и CSS (англ. Cascading Style Sheets -— каскадные таблицы стилей), а также использования готовых наборов для создания веб-интерфейсов, таких как Twitter Bootstrap. В качестве основной программной архитектуры был выбран набор шаблонов проектирования MVC (англ. Model-view-controller — модель-представление-контроллер), с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Схема взаимодействия между отдельными частями системы представлена на рисунке 2.

Диаграмма компонентов представлена на рисунке 3.

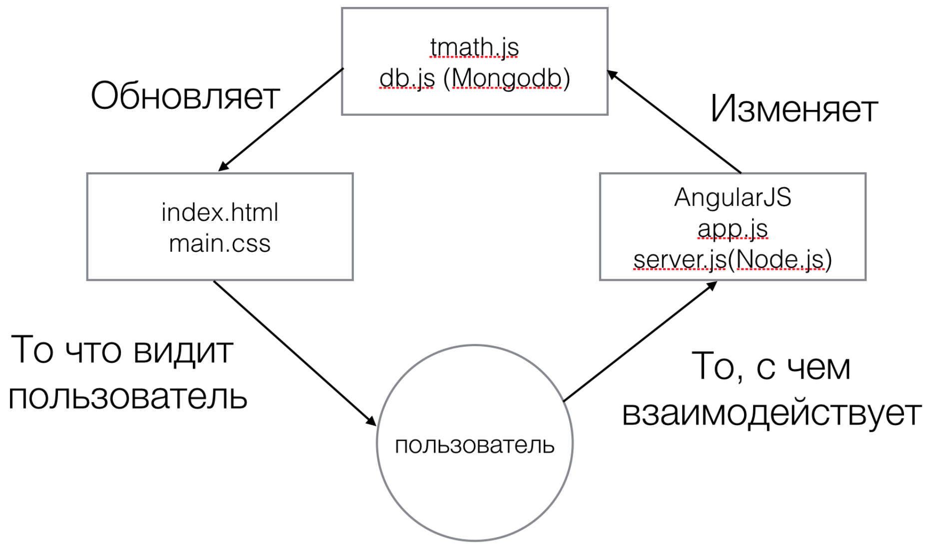

Рис. 2: Струтура веб-приложения

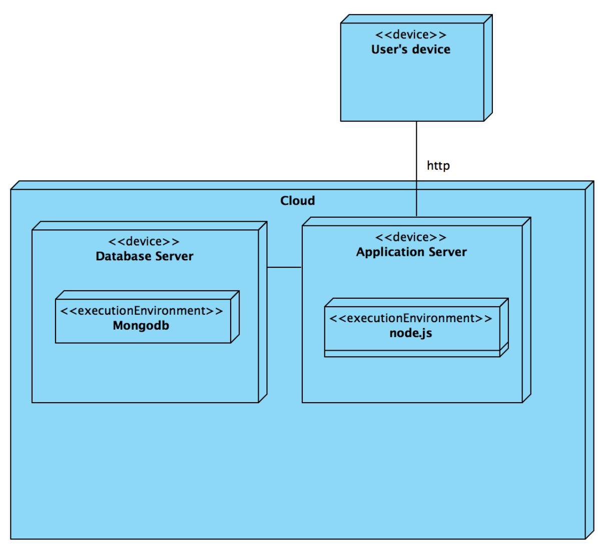

Рис. 3: Диаграмма компонентов

3.2 Описание программного и технического обеспечения

Основной мотив выбора языка JavaScript был частично описан ранее и состоит в его распространенности, гибкости, возможности использовать его как в клиентской, так и в серверной части приложения. Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса. JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам — функции как объекты первого класса, объекты как списки, карринг, анонимные функции, замыкания, что придаёт языку дополнительную техническую гибкость. В качестве тестирующего инструмента был выбран фреймверк Jasmine, каждый компонент разработанной библиотеки был покрыт несколькими позитивными тестами. В качестве базы данных приложения была выбрана MongoDB — документо-ориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. Данный инструмент имеет следующие возможности:

- Документо-ориентированное хранение (англ. JSON — JavaScript Object Notation-подобная схема данных)
- Динамические запросы
- Поддержка индексов
- Профилирование запросов
- Быстрые обновления «на месте»

- Поддержка отказоустойчивости и масштабируемости: асинхронная репликация, набор реплик и распределения базы данных на узлы

- Работа в соответствии с парадигмой MapReduce

В качестве сервера был выбран Node.js — программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Главные преимущества использования Node.js в предложеном решении:

- Единый язык разработки(JavaScript)

- Асинхронность

- Событийность

- Масштабируемость

В качестве основного фреймверка был выбран AngularJS, который предназначен для разработки одностраничных приложений. Его цель — расширение браузерных приложений на основе MVC шаблона, а также упрощение тестирования и разработки. Фреймворк работает с HTML, содержащим дополнительные пользовательские атрибуты, которые описываются директивами, и связывает ввод или вывод области страницы с моделью, представляющей собой обычные переменные JavaScript. Значения этих переменных задаются вручную или извлекаются из статических или динамических JSON-данных. Выбор был обусловлен следующими приемуществами:

- Отделение DOM (англ. Document Object Model –– объектная модель документа) манипуляции от логики приложения, что улучшает тестируемость кода.

- Отношение к тестированию как к важной части разработки. Сложность тестирования напрямую зависит от структурированности кода.

- Разделение клиентской и серверной стороны, что позволяет вести разработку параллельно.

Возможность дальнейшего развития проекта облегчена путем использования облачных инструментов разработки от Cloud9 и распределенной системы управления версиями файлов — Git, что позволило снизить порог вхождения из-за упрощенной первичной настройки и сборки проекта, использовать единый инструментарий в разработке и может облегчить взаимодействие из-за наличия возможности редактирования кода несколькими программистами одновременно.

3.3 Руководство пользователя

Пользователь может зайти в систему, используя логин и пароль, или в качестве гостя, в таком случае, вводить пароль не потребуется. Окно со списком уже ранее созданных проектов представлено на рисунке 4. Пользователь может удалять, редактировать и добавлять новые проекты, создание нового проекта изображено на рисунке 5. Если пользователь ввел новое имя проекта, то при нажатии на кнопку справа от ввода имени, новый проект добавится в список проектов, как это изображено на рисунке 6. После нажатия на конкретный проект, пользователь попадет в окно проекта, где ему будет предложено создать новую задачу — рисунок 7. После ввода имени и нажатия

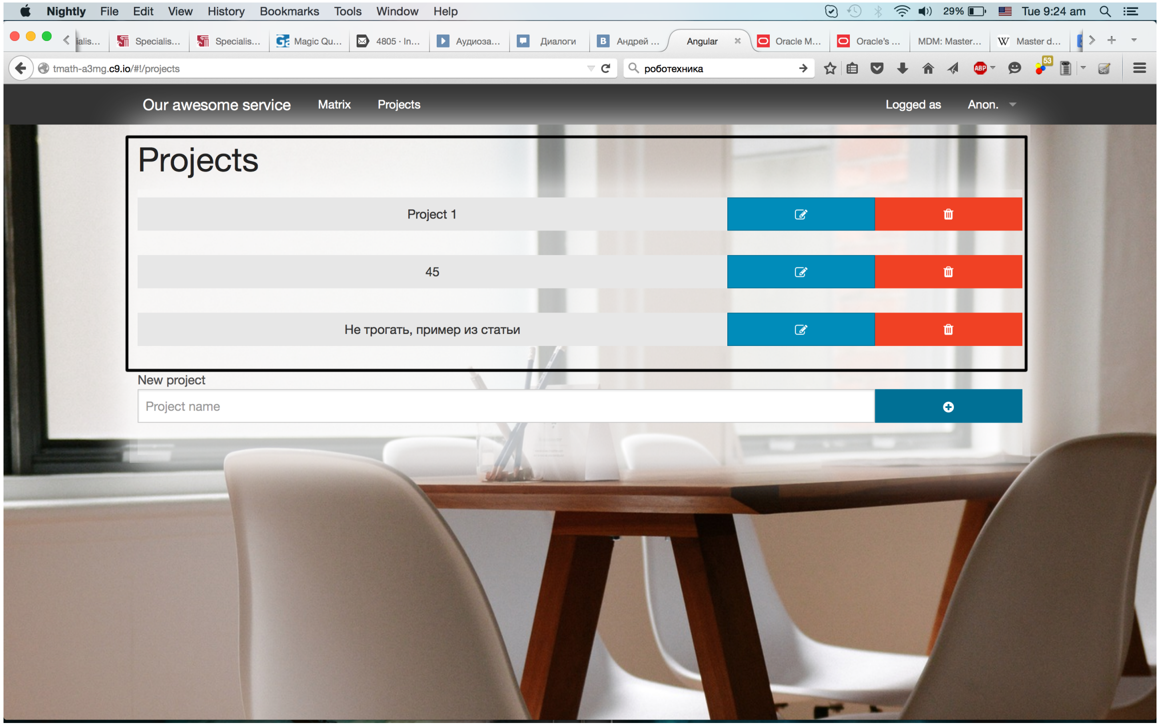

Рис. 4: Список текущих проектов

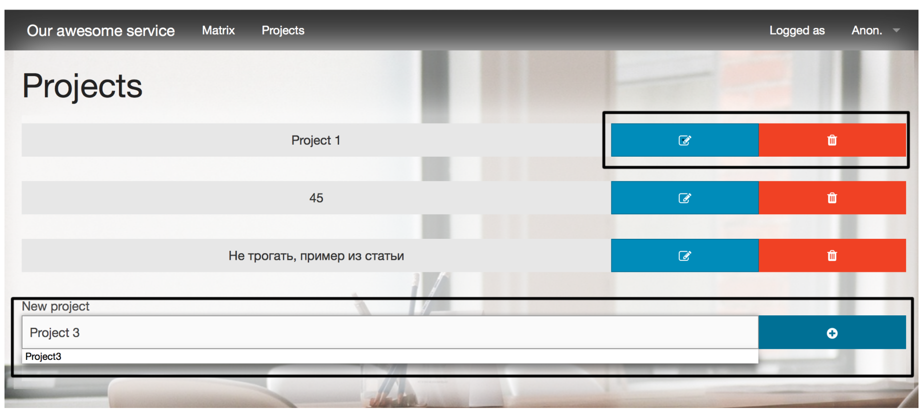

Рис. 5: Добавление нового проекта

на кнопку справа от ввода имени, пользователю станет доступно редактирование задачи, как это показано на рисунке 8. Добавление связи к отдельной задаче представлено на рисунке 9. Пользователю доступно добавление неограниченного количества связей между уже созданными задачами. Сохранение данных происходит после нажатия на кнопку снизу или на кнопку закрытия, которая находится в правом верхнем углу. Редактирование связи представле-

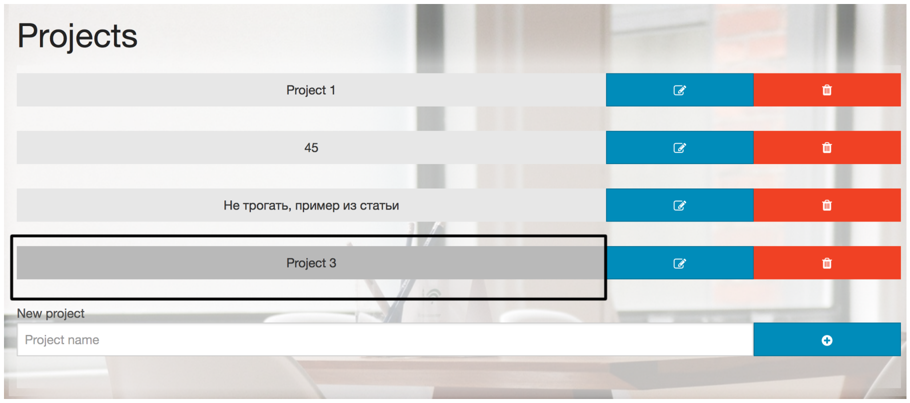

Рис. 6: Проект успешно добавлен

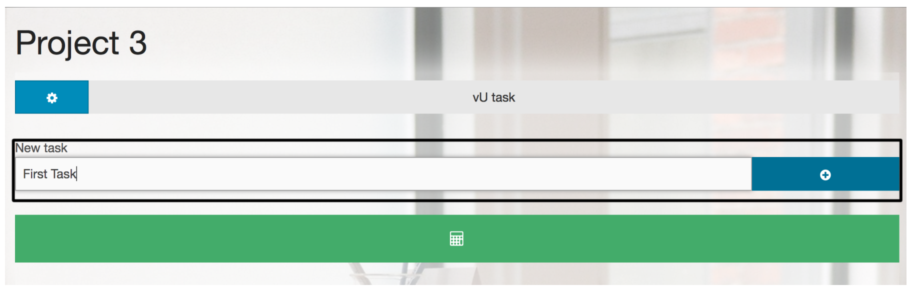

Рис. 7: Добавление новой задачи

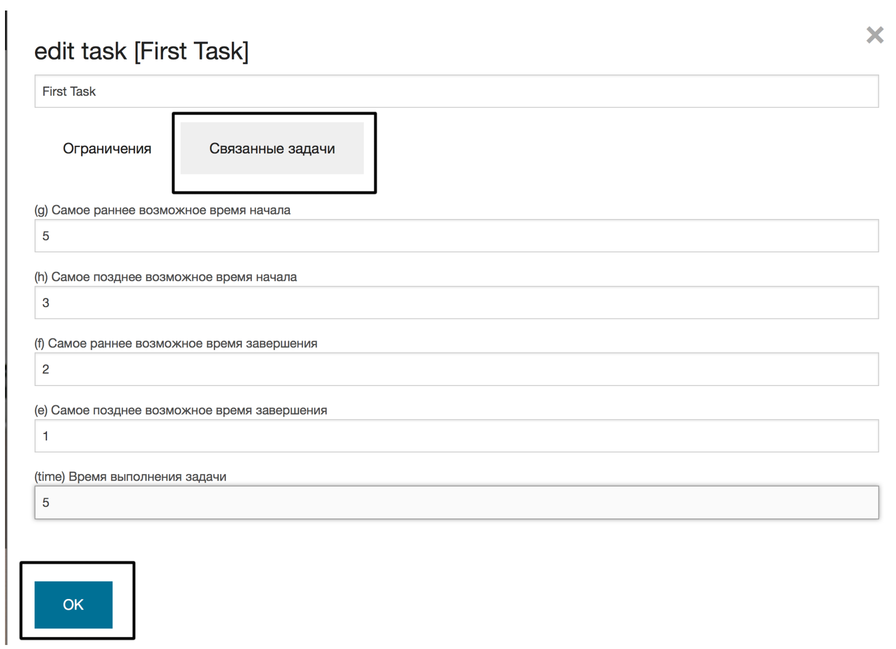

Рис. 8: Заполнение параметров, переход к связям задачи

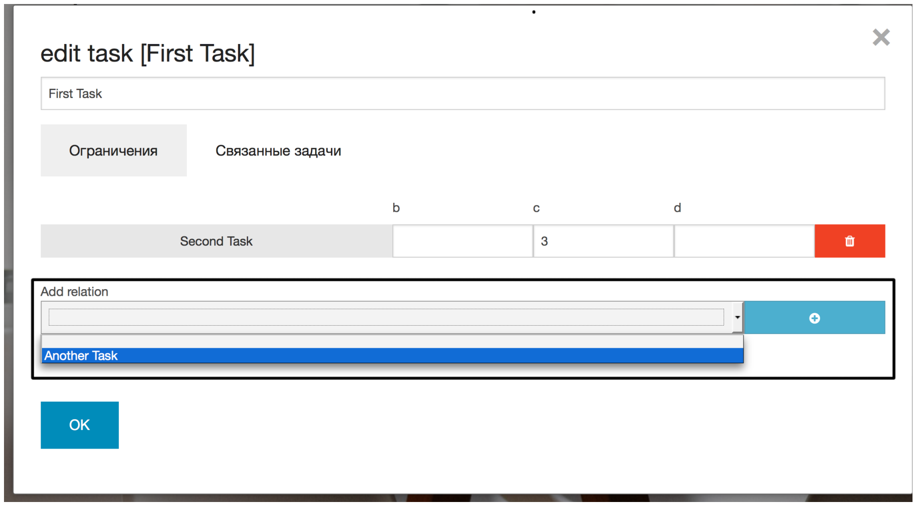

Рис. 9: Добавление связи задачи

но на рисунке 10, все механизмы создания и редактирования аналогичны как для задач, так для проектов и связей, за исключением того, что связь нужно предварительно выбрать в выпадающем списке. При добавлении нужного

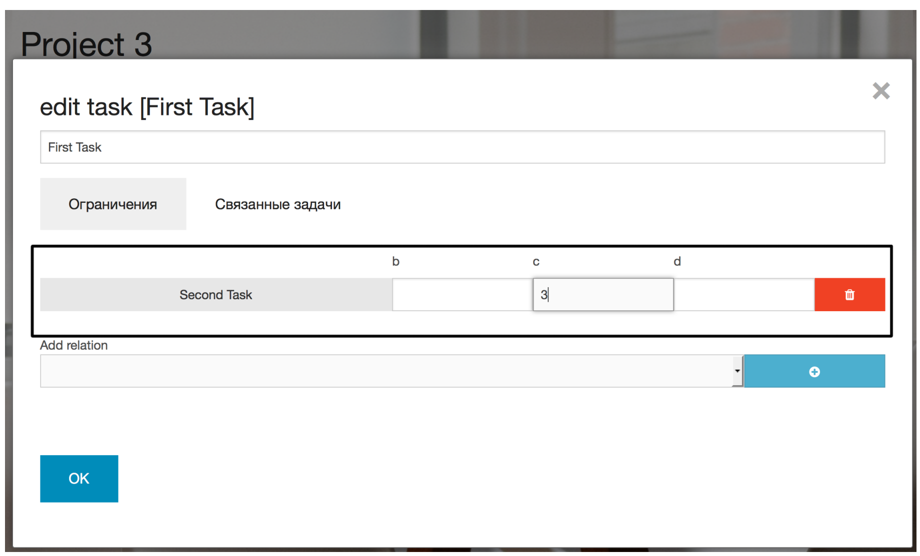

Рис. 10: Редактирование связи

количества задач и заполнения связей и ограничений пользователь может нажать на кнопку внизу окна проекта, пример составления представлен на рисунке 11.

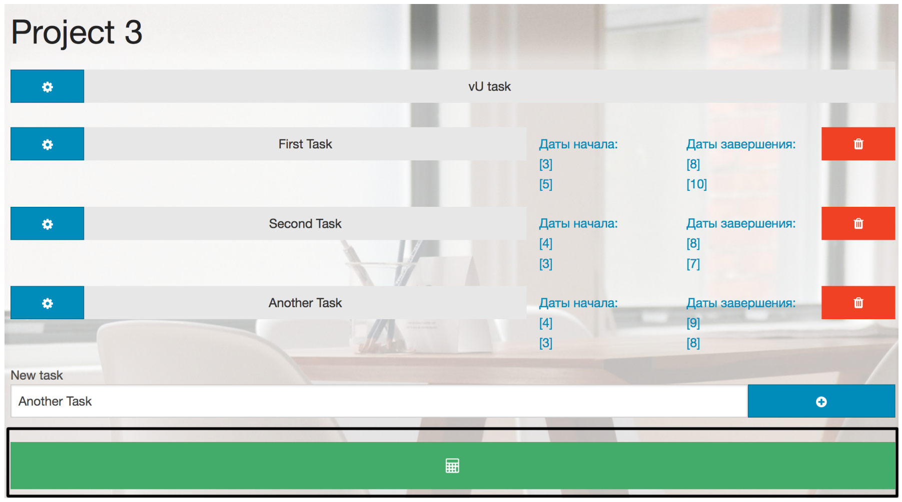

Рис. 11: Составление расписания

Заключение

В результате выполнения магистерской диссертации была изучена предметная область управления проектами и стандартные подходы к решению задач сетевого планирования. Также были изучены существующие программные средства управления проектами и методы решения задач планирования с помощью аппарата идемпотентной математики. Было разработано комплексное программное решение состоящее из следующих компонентов: библиотеки идемпотентной математики (JavaScript), сервера, обрабатывающего запросы от пользователей (Node.js) и базы данных (MongoDB), хранящей результаты работы алгоритмов. Все вычисления и библиотека покрыты тестами (Jasmine) для проверки корректности работы программного решения.

Список литературы

1. R. A. Cuninghame-Green, “Describing industrial processes with interference and approximating their steady-state behaviour,” *Oper. Res. Quart.* **13** no. 1, (1962) 95–100. `http://www.jstor.org/stable/3007584`.

2. Руководство к Своду знаний по управлению проектами. Пятое издание (Руководство PMBOK) Американский национальный стандарт ISBN 978-1-62825-008-4 — Fifth edition. — Project Management Institute, 2013. — P. 380.

3. N. Krivulin, Tropical optimization problems with application to project scheduling with minimum makespan, arXiv:1403.0268 , 2015

4. N. Krivulin, Tropical optimization problems in project scheduling arXiv:1502.06222 , 2015

5. J. S. Golan, *Semirings and Affine Equations Over Them: Theory and Applications*, vol. 556 of *Mathematics and Its Applications.* Kluwer Academic Publishers, Dordrecht, 2003.

6. B. Heidergott, G. J. Olsder, and J. van der Woude, *Max-plus at Work: Modeling and Analysis of Synchronized Systems*. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, 2006.

7. M. Gondran and M. Minoux, *Graphs, Dioids and Semirings: New Models and Algorithms*, vol. 41 of *Operations Research / Computer Science Interfaces.* Springer, New York, 2008.

8. P. Butkovič, *Max-linear Systems: Theory and Algorithms.* Springer Monographs in Mathematics. Springer, London, 2010.

9. W. M. McEneaney, *Max-Plus Methods for Nonlinear Control and Estimation*. Systems and Control: Foundations and Applications. Birkhäuser, Boston, 2006.

10. K. Zimmermann, "Interval linear systems and optimization problems over max-algebras," in *Linear Optimization Problems with Inexact Data*, pp. 165–193. Springer, New York, 2006.

11. P. Butkovič and A. Aminu, "Introduction to max-linear programming," *IMA J. Manag. Math.* **20** no. 3, (2009) 233–249. `http://imaman.oxfordjournals.org/content/20/3/233.full.pdf+html`.

12. P. Butkovič and K. P. Tam, "On some properties of the image set of a max-linear mapping," in *Tropical and Idempotent Mathematics*, G. L. Litvinov and S. N. Sergeev, eds., vol. 495 of *Contemp. Math.*, pp. 115–126. American Mathematical Society, 2009.

13. A. Aminu and P. Butkovič, "Non-linear programs with max-linear constraints: A heuristic approach," *IMA J. Manag. Math.* **23** no. 1, (2012) 41–66. `http://imaman.oxfordjournals.org/content/23/1/41.full.pdf+html`.

14. N. Krivulin, "A maximization problem in tropical mathematics: A complete solution and application examples," `arXiv:1304.7461 [math.OC]`.

15. N. Krivulin, "A multidimensional tropical optimization problem with nonlinear objective function and linear constraints," *Optimization* (2013) , `arXiv:1303.0542 [math.OC]`.

16. N. Krivulin, "Explicit solution of a tropical optimization problem with application to project scheduling," in *Mathematical Methods and Optimization Techniques in Engineering*, D. Biolek, H. Walter, I. Utu, and C. von Lucken, eds.,

pp. 39–45. WSEAS Press, 2013. arXiv:1303.5457 [math.OC]. http://www.wseas.us/e-library/conferences/2013/Antalya/OTEMA/OTEMA-03.pdf.

17. N. Krivulin, "Extremal properties of tropical eigenvalues and solutions to tropical optimization problems," *Linear Algebra Appl.* (2014) , arXiv:1311.0442 [math.OC].

18. N. Krivulin, "A constrained tropical optimization problem: Complete solution and application example," in *Tropical and Idempotent Mathematics and Applications*, G. L. Litvinov and S. N. Sergeev, eds., vol. 616 of *Contemp. Math.*, pp. 163–177. AMS, Providence, RI, 2014. arXiv:1305.1454 [math.OC].

19. E. L. Demeulemeester and W. S. Herroelen, *Project Scheduling: A Research Handbook*. International Series in Operations Research and Management Science. Springer, 2002.

20. V. T'kindt and J.-C. Billaut, *Multicriteria Scheduling: Theory, Models and Algorithms*. Springer, Berlin, 2006.

21. M. Vanhoucke, *Project Management with Dynamic Scheduling*. Springer, Berlin, 2013.

22. Н. К. Кривулин, *Методы идемпотентной алгебры в задачах моделирования и анализа комплексных систем*. Издательский Дом Санкт-Петербургского ниверситета, Санкт-Петербург, 2009. http://www.google.ru/books?id=PDQP7kIGrhMC.