ПРИМЕНЕНИЕ АЛГОРИТМА СЕТЕВОГО ПЛАНИРОВАНИЯ К ЗАДАЧАМ УПРАВЛЕНИЯ ПРОЕКТОМ

О. В. Зданевич

Белорусский государственный университет, г. Минск;
oleg.zdanevich@icloud.com;
науч. рук. – О. И. Пиндрик, канд. физ.-мат. наук, доц.

Для качественного планирования и анализа проектов используется инструментарий для построения сетевых моделей, которые способны объяснить сложную структуру взаимосвязей между задачами проекта, а также, в свою очередь, являются инструментом поддержки принятия решений для руководителя проекта. Однако, большинство специализированных средств для управления проектами не имеют возможности расчета сетевых моделей и определения сроков начала задач, а те, что имеют обходятся слишком дорого или имеют данный функционал в «зачаточном» состоянии.

Основным результатом данной работы является веб-приложение для расчета ресурсного плана с целью оптимизации времени выполнения проекта. Результат работы представлен в виде диаграммы Гантта.

Ключевые слова: исследование операций; сетевые модели; веб-разработка; диаграммы Гантта; Java; Spring; React.

Методы сетевого планирования используются при реализации весьма трудоемких и значимых проектов. Сфера применения сетевого планирования весьма обширна:

1. Научные исследования.
2. Сельскохозяйственные и производственные работы.
3. Инновационная деятельность.
4. Бизнес планирование и прочее.

Проведение подобного рода проектов требует определенной календарной увязки большого числа взаимосвязанных работ, которые выполняются различными исполнителями. Планирование, анализ и оптимизация, при немалом потоке информации, являются достаточно сложными задачами. Как правило, руководителю сложно «держать в голове» все что нужно для того, чтобы с минимальной погрешностью принимать управленческие решения. В этом случае на помощь приходят методы сетевого планирования.

Сетевой моделью (СМ) называется графическое представление плана выполнения комплекса работ (проекта), заданного в специфической форме – в виде сети. Графическое изображение подобной сети называется сетевым графиком (рис.1).

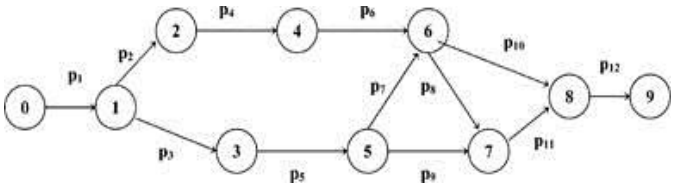

Рис. 1. Пример сетевой модели

Оптимизация сетевого графика заключается в усовершенствовании планирования временных, экономических ресурсов и, как следствие, сокращение затрат на выполнение проекта.

Пусть проект, который будет в дальнейшем оптимизирован, состоит сразу из нескольких подпроектов. Каждый подпроект выполняется независимо и представлен отдельной сетевой моделью, но связанных с другими подпроектами глобальными ресурсами (рис.2).

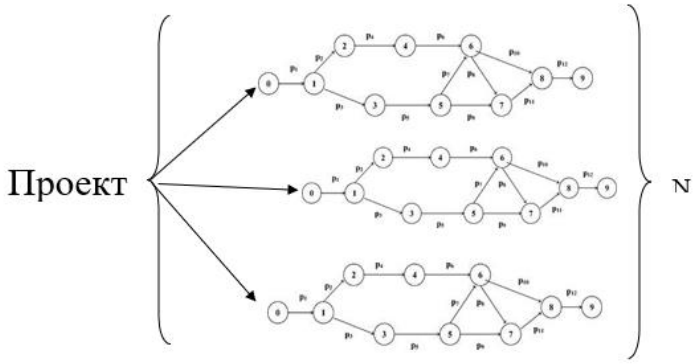

Рис. 2. Пример проекта

Ресурс – это именованная единица, которая необходима для выполнения работ (каждая работа может требовать n-ресурсов в количестве m единиц). Ресурсы бывают нескольких типов:

1. Глобальный или Локальный.
2. Восстановимый или не восстановимый.

Также могут существовать разные **режимы выполнения** для каждой из задач, те разные способы выполнения одной и той же работы могут требовать:

1. Различное время выполнения.
2. Различные наименования ресурсов.
3. Различное количество ресурсов.

В данной работе рассматриваемый проект (рис. 2) будет оптимизирован по следующим критериям:

1. Ресурсы должны быть использованы так, чтобы в любую единицу времени, количество свободных ресурсов было больше либо равно, количеству ресурсов необходимых для продолжения проекта.

2. Время конца выполнения отдельных сетевых моделей должно быть минимальным.

3. Время конца выполнения всего проекта должно быть минимальным.

Задача оптимизации решена с помощью алгоритма «**FirstFit**» который обычно используется для решения «задач об упаковке».

Пошаговая работа алгоритма «FirstFit»:

1. Перебирать предметы по порядку.
2. Для каждого предмета просматривать ящики по порядку (по возрастанию номеров).
3. Если найден подходящий ящик, то разместить предмет в этом ящике. Подходящий – это тот, в котором можно разместить текущий предмет.
4. Если подходящего ящика нет – создаём новый и размещаем предмет в нём.

Для того, чтобы применить решение «задачи об упаковке» для сетевых моделей необходимо переопределить несколько понятий.

Пусть существует m ресурсов, и существует пространство $R^M \mathrm{R}^{\mathrm{M}}$, тогда $\exists\vec{\alpha} = (\alpha_1, \dots, \alpha_m)$, который для каждой задачи, в любую единицу времени обозначает потребность этой задачи в ресурсах.

В терминах «задачи об упаковке», это будет размер одной вещи.

Пусть так же:

- в любую единицу времени у нас $\exists\vec{\beta} = (\beta_1, \dots, \beta_{\mathrm{m}})$, который обозначает свободное количество ресурсов, в любую единицу времени;
- $\exists(\gamma_1, \gamma_2) \in \mathrm{Z}^1$, который является интервалом времени, в нашей задаче;
- $\exists \mathrm{B} = \{\vec{\beta}_{\mathrm{i}}\}, \mathrm{i} \in [\gamma_1, \gamma_2] \in \mathrm{Z}^1$, это множество в терминах задачи об упаковке является одной упаковкой.

Тогда мы говорим, что предмет помещается в упаковку в случае, если $\forall\vec{\beta_i} \in B$, выполняется условие $(\beta_i - \alpha_i) \geq 0, \forall i \in [1, m] \in Z^1$.

При этомпроцесс создания новой коробки — это найти такие $[\gamma_1, \gamma_2] \in Z^1$, для которых будет выполняться условие, что предмет помещается в коробку, и они будут минимальны.

Все понятия алгоритма «First fit» как и задачи о коробки, могут быть представлены в задаче, которая была обозначена, в данной работе, от сюда следует, что, мы можем решать задачу о ресурсах в сетевой модели, как «задачу об упаковке».

Далее будем рассматривать само **веб-приложение.**

Технологический стек данного приложения:

Средства контейнеризации:

1. Docker – средство контейнеризации, необходимое для управления контейнерами, внутри которых находятся отдельные сервисы.

База Данных:

1. MySQL 8 – СУБД, которое используется для хранения данных о проектах, а также о пользователях.

Серверная часть:

1. Java 8 – основной язык разработки.
2. Maven – фреймворк для автоматической сборки и управления пакетами.
3. SpringFramework.
    - 3.1. SpringBoot – средство для ускорения разработки и настройки сервера Tomcat в данном случае.
    - 3.2. SpringSecurity – технология для управления безопасностью всего приложения.
    - 3.3. Spring Data JPA – более высокоуровневая технология для работы с СУБД, внутри которой используется Hibernate.
    - 3.4. Spring Web MVC – фреймворк для реализации MVC концепции
4. Hibernate – ORM технология, используемая для работы с СУБД.

Клиентская часть:

1. HTML – язык разметки.
2. CSS – язык стилей.
3. JavaScript – основной язык программирования на стороне клиента.
4. React – JS фреймворк на котором написан весь пользовательский интерфейс.
5. Bootstrap – набор инструментов оформления пользовательского интерфейса.
6. React-modules – набор дополнительных модулей для React.
7. NPM – менеджер пакетов.
8. Nginx – веб-сервер, на котором размещается клиентская часть.

Для того чтобы воспользоваться приложением, нужно зарегистрироваться и войти в него. После входа необходимо создать новый проект и задать: количество моделей, ресурсы, задания внутри моделей, а также, для каждого из заданий указать режимы выполнений (возможные варианты связки «ресурсы – время выполнения задания данным множеством ресурсов»). Сразу после создания, проект будет оставаться еще не просчитанным. Для просчета проекта необходимо нажать на кнопку «Посчитать». После того как модель будет просчитана, отобразиться диаграмма Гантта, в которой разными цветами будут обозначаться разные модели входящие в проект (рис 3).

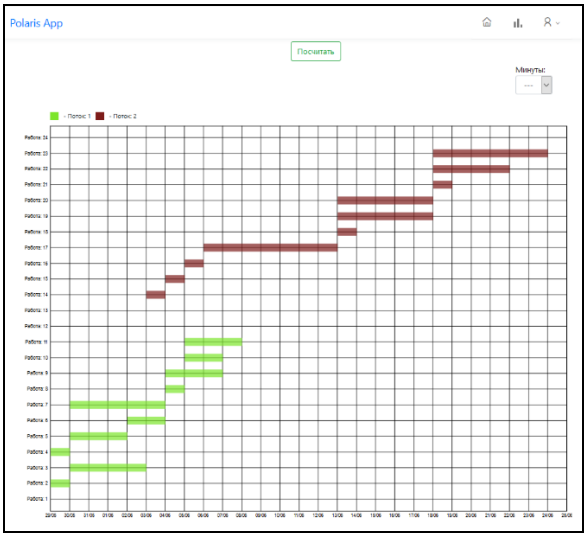

Рис. 3. Результат работы приложения – диаграмма Гантта

Итогом данной работы, является самостоятельное веб-приложение, которое, используя методы сетевого планирования, решает задачу по распределению ресурсов, а также проводит оптимизацию времени длительности выполнения проекта. Данное приложения может применяться в организациях, в которых возможно долговременное планирование такие, как, например, университеты, министерства и другие учреждения различных форм собственности. Это приложение является практикоориентированными позволяет облегчить работу служащих, а также избежать ошибок планирования, которые, в свою очередь, могут повлечь потерю времени и финансов предприятия.

Библиографические ссылки

1. Исследование операций : курс лекций / В. И. Бахтин [и др.]. Мн., 2003.
2. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М., 1982.
3. *Рассел Д., Кон Р.* Диаграмма Гантта. VSD, 2012.